Keynote Paper

Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives

Radu Marculescu, *Senior Member, IEEE*, Umit Y. Ogras, *Student Member, IEEE*, Li-Shiuan Peh, Natalie Enright Jerger, and Yatin Hoskote, *Senior Member, IEEE*

***Abstract*—To alleviate the complex communication problems that arise as the number of on-chip components increases, network-on-chip (NoC) architectures have been recently proposed to replace global interconnects. In this paper, we first provide a general description of NoC architectures and applications. Then, we enumerate several related research problems organized under five main categories: Application characterization, communication paradigm, communication infrastructure, analysis, and solution evaluation. Motivation, problem description, proposed approaches, and open issues are discussed for each problem from system, microarchitecture, and circuit perspectives. Finally, we address the interactions among these research problems and put the NoC design process into perspective.**

***Index Terms*—Energy and power consumption, multiprocessor systems-on-chip (MPSoCs), networks-on-chip (NoCs), on-chip communication.**

I. Introduction

TRADITIONALLY, the design space exploration for systems-on-chip (SoCs) has focused on the computational aspects of the problem at hand. However, as the number of components on a single chip and their performance continue to increase, the design of the communication architecture plays a major role in defining the area, performance, and energy consumption of the overall system. Furthermore, with technology scaling, the global interconnects cause severe on-chip synchronization errors, unpredictable delays, and high power consumption [64]. To mitigate these effects, the network-on-chip (NoC) approach emerged recently as a promising alternative to classical bus-based and point-to-point (P2P) communication architectures [40], [74], [107]. Aside from better predictability and lower power consumption, the NoC approach offers greater scalability compared to previous solutions for on-chip communication.

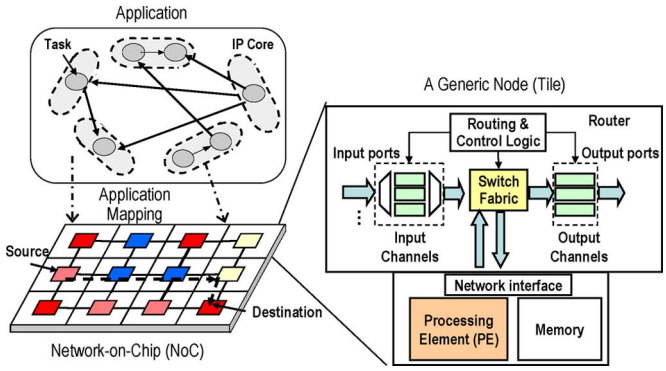


Fig. 1. Generic NoC architecture, application, and its mapping to the NoC are shown. The anatomy of a node which consists of an on-chip router, buffers, and processing element (PE) is also shown on the right-hand side of the figure.

As shown in Fig. 1, modern SoC architectures consist of heterogeneous IP cores such as CPU or DSP modules, video processors, embedded memory blocks, etc. Each such processing element (PE) is attached to a local router which connects the core to the neighboring nodes via a NoC. More precisely, when a source node sends a packet to a destination node (see Fig. 1), the packet is first generated and transmitted from the local processor to the attached router via a network interface (NI). The NI enables seamless communication between various cores and the network. Then, the packet is stored at the input channels, and the router starts servicing it. This service time includes the time needed to make a routing decision, allocate a channel, and traverse the switch fabric. After being serviced, the packet moves to the next router on its path, and the process repeats until the packet arrives at its final destination. As a result, the communication among various cores is achieved by generating, processing, and forwarding packets through the network infrastructure rather than by routing global wires. Not surprisingly, the network communication latency depends on the characteristics of the target application (e.g., inter-task communication volume), computational elements (e.g., processor speed), and network characteristics (e.g., network bandwidth and buffer size).

Manuscript received February 27, 2008; revised June 3, 2008 and September 2, 2008. This work was supported in part by Marco Gigascale Systems Research Center, one of the five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation (SRC) program, under SRC Contracts 2008-HJ-1823 and 2008-HJ-1800 and in part by the National Science Foundation under Grant CCF-0702420. This paper was recommended by Associate Editor G. E. Martin.

R. Marculescu is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: radum@ece.cmu.edu).

U. Y. Ogras was with Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Strategic CAD Laboratories, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: umit.y.ogras@intel.com).

L.-S. Peh is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: peh@princeton.edu).

N. E. Jerger was with the Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI 53706 USA. She is now with the University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: enrightn@cae.wisc.edu).

Y. Hoskote is with the Corporate Technology Group, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: yatin.hoskote@intel.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2008.2010691

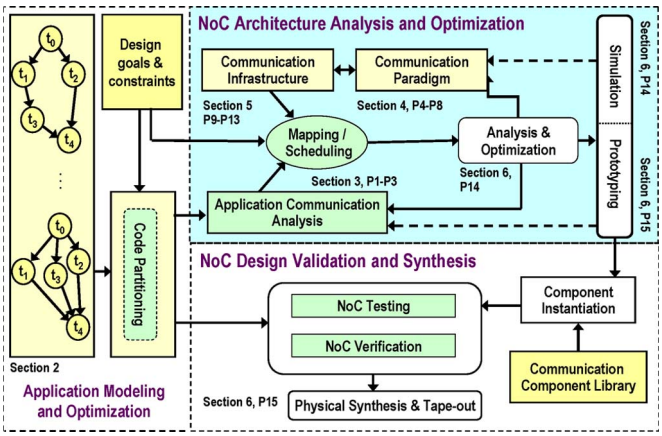


Fig. 2. Generic NoC synthesis flow consisting of 1) application specification and modeling, 2) NoC communication architecture optimization, and 3) NoC testing and verification. Various problems P1–P15 are also shown.

Developing a design methodology for NoC-based communication poses novel and exciting challenges to the electronic design automation (EDA) community. While a handful of design problems have been recently identified by several researchers [21], [62], [120], a structured presentation of the major research themes is still missing. Having such a unifying presentation available can not only catalyze research toward improving the existing solutions, but also inspire new solutions to the many outstanding research problems in NoC design. Hence, rather than simply surveying prior work, the objective of this paper is to clarify the fundamental issues in NoC design and, more importantly, provide a roadmap for future research. From this perspective, this paper focuses on the NoC design methodologies and computer-aided design algorithms aimed at system, microarchitecture, and circuit levels of abstraction.

From a design methodology perspective, many design choices about the on-chip interconnect need to be made. The factors affecting these choices range from application and platform decisions, all the way down to the circuit- and layout-level considerations (see Fig. 2). Here, we first address some of the difficult choices that the NoC designers face, before presenting them as broad categories open for research. More precisely, we consider explicitly the following research issues:

1) **Application Modeling and Optimization.** One of the first questions to be answered is what are the target applications and their associated traffic patterns, as well as the interconnect bandwidth requirements for each node in the network (see leftmost box in Fig. 2). In coherent shared memory architectures, the communication patterns depend primarily on the flow of external memory traffic and the on-die cache hierarchy and coherence protocol. On the other hand, noncoherent shared memory or message-passing models depend more directly on the explicit communication patterns of various applications. Clearly, the application partitioning and overall system architecture (e.g., homogenous versus heterogeneous cores, synchronous versus asynchronous clocking, memory controllers, I/O devices, etc.) significantly impact the network traffic. Having a good model for the application helps find the best application–architecture combinations that satisfy various performance and energy constraints.
2) **NoC Communication Architecture Analysis and Optimization.** A good understanding of the traffic patterns and system requirements helps determine the optimal network topology; this has a huge impact on design costs, power, and performance, and helps designers choose an efficient routing algorithm and flow control scheme in order to manage the incoming traffic. These issues are shown in the upper right box in Fig. 2 and are discussed in problems P4–P13 later in this paper. At this point, communication bandwidth and network latency are the key performance metrics, while area, power, and reliability are the key cost metrics.
3) **NoC Communication Architecture Evaluation.** Having an initial NoC design, a performance model is needed to identify the congestion points, hot spots, or other such issues (see the Analysis and Optimization box and P14 in Fig. 2). With confidence gained through analysis and simulation, the designer can quickly proceed with circuit and backend design (problem P15 in Fig. 2). In addition to performance, power models are crucial. Early and efficient floorplanning and accurate estimation of the area and power are necessary for optimizing the design and ensuring a quick convergence of the backend design flow.
4) **NoC Design Validation and Synthesis.** Finally, the chosen NoC design can be prototyped, validated, and tested, as shown in the lower right box in Fig. 2. Prototyping, verification, and testing are also used to address various resilience and fault tolerance issues early on.

In summary, the NoC design is a multifaceted process involving both application and communication architecture. In [21], the authors provide a comprehensive introduction to NoCs and existing design practices. Instead, this paper focuses on understanding and structuring the main research problems in NoC design via problem motivation, description, proposed solutions, and open issues. The emphasis of this paper is on the algorithmic optimization of the communication architecture under various energy and performance constraints, while programmability and software aspects are left for future work.

The remainder of this paper is organized as follows. Section II introduces a formalism needed for application and architecture description. Several outstanding research issues related to application optimization, communication paradigm, and communication infrastructure optimization are addressed in Sections III–V, respectively. Section VI discusses various analysis, simulation, and prototyping approaches. Finally, the interactions among these research ideas and likely trends in NoC design are discussed in Section VII.

II. Application and Architecture Descriptions

We first present a few definitions regarding the target application and NoC architecture that will be used later in this paper.

Definition 1: An application characterization graph (APCG) $G = G(C, A)$ is a *directed* graph, where each vertex $c_i \in C$ represents an IP core and each directed arc $a_{i,j} \in A$ characterizes the communication from vertex c_i to vertex c_j. Each $a_{i,j}$ can be tagged with application-specific information (e.g., communication volume $v(a_{i,j})$ between vertices c_i and c_j)

and specific design constraints (e.g., communication bandwidth $b(a_{i,j})$ and latency requirements application, etc.). Of note, for simplicity reasons, this definition assumes implicitly that application tasks are already bound to the IP cores of the platform (see Fig. 1).

Definition 2: An NoC architecture can be uniquely described by the triple $\text{Arch}(T(R,Ch), P_R, \Omega(C))$, where

1) The labeled graph $T(R,Ch)$ represents the network *topology*. The routers and channels in the network are given by the sets R and Ch, respectively, as follows:
   a) $\forall(ch) \in Ch$, $w(ch)$ gives the bandwidth of channel ch.
   b) $\forall r \in R$, $l(ch,r)$ gives the buffer size (depth) of channel ch, located at router r.
   c) $\forall r \in R$, $\text{Pos}(r)$ gives the xy coordinates of router r in the chip floorplan.
2) $\{P_R(r,i,j)|i,j,r \in R\}$ defines the routing policy P_R at router r, for any source router i and destination router j, while considering a particular switching technique.
3) $\Omega : C \rightarrow R$ is a function that maps each vertex $c_i \in C$ in the APCG to a router in R. For *direct* topologies, Ω is a bijective function, i.e., every router is connected to a core, while in *indirect* topologies, a router may be connected only to other routers.

We can conceive the choices of designing NoCs as representing a 3-D design space, where each component of the triple $\text{Arch}(T(R,Ch), P_R, \Omega(C))$ defines a separate dimension of the design space.[1] Using this type of concise representation has the advantage of keeping the discussion simple and precise.

Finally, besides the obvious functional constraints like correctness, freedom from deadlock, etc., there are a number of *performance* and *cost* metrics that the design techniques developed for NoCs should comply with. The following is a non-exhaustive list of such metrics

$$\begin{aligned}&\text{Performance Metrics}\\&\quad=\{\,\text{average/maximum packet latency},\\&\qquad\text{bisection bandwidth},\\&\qquad\text{network throughput, QoS}\,\}\end{aligned} \tag{1}$$

$$\begin{aligned}&\text{Cost Metrics}\\&\quad=\{\,\text{average/peak energy/power consumption},\\&\qquad\text{network area overhead, total area},\\&\qquad\text{average/peak temperature}\,\}.\end{aligned} \tag{2}$$

The optimization goals of various research problems discussed hereafter and the corresponding design constraints reflect different metrics belonging to these sets. These metrics can be used as an objective function $O(\text{Arch}, G)$ or constraints $\text{Const}(\text{Arch}, G)$ expressed as a function of the architecture Arch and application parameters G, respectively. For example, (5), given later in this paper, uses the communication energy consumption in (2) as the cost function and expresses it as a function of the target application [through communication volume $v(a_{i,j})$] and the NoC architecture (through mapping function Ω and energy E_{bit} required to move 1 b of data between any two cores c_i and c_j).

[1]We note that similar definitions have been used by the EDA research community [60], [70], [115], [159].

III. Application Modeling and Optimization for NoC Communication

Using the definitions in Section II, we describe the first set of research problems (P1–P3) for NoC application modeling and optimization. Generally speaking, traffic modeling and benchmarking help designers in determining effective architectures, while application mapping and scheduling are means to optimize the NoC performance and/or energy consumption.

A. P1 Traffic Modeling and Benchmarking

1) Motivation: Traffic models refer to the mathematical characterization of workloads generated by various classes of applications. With network performance being highly dependent on the actual traffic, it is obvious that accurate traffic models are needed for a thorough understanding of the huge design space of network topologies, protocols, and implementations. Since implementing real applications is time consuming and lacks flexibility, analytical models can be used instead to evaluate the network performance early in the design process.

2) Problem Description: Let $X(t)$, $t \in Z^+$, be a stochastic process denoting the traffic volume (bits, packets, etc.) at time instance t. The mean $(E[X])$, the second-order statistics such as variance $(E[(X - E[X])^2])$, and the autocorrelation function $R(k)$ reveal important information about the time series that capture the actual traffic characteristics.

Time series were first analyzed with autoregressive moving average models (i.e., Markovian or short-range dependent models), where the autocorrelation function decays exponentially fast; i.e.,

$$R(k) \sim e^{-\beta k}, \qquad \text{as } k \rightarrow \infty \tag{3}$$

where β is a positive constant that shows the rate of decay and "$\sim$" denotes the "*asymptotically close*" condition. Note that, for short-range models $\Sigma_{k=0}^{\infty} R(k)$ is finite. However, due to the complex nature of applications and intrinsic properties of network protocols, the vast majority of time series describing the network traffic have an autocorrelation function that decays at a slower rate compared to the exponential function; this property defines the so-called *long-range* dependent traffic

$$R(k) \sim k^{-\alpha}, \qquad \text{as } k \rightarrow \infty \tag{4}$$

where $\alpha \in [0,1]$. Of note, for long-range dependent processes, the sum $\Sigma_{k=0}^{\infty} R(k)$ diverges. A special class of long-memory processes is represented by the so-called self-similar traffic which is characterized by the Hurst parameter (H). The Hurst parameter satisfies $H \in [0.5, 1)$ which is the characteristic of self-similarity [130]. Using such stochastic models, the traffic modeling problem is formulated as follows:

Given the traces or workload characterization of the target applications (e.g., packet injection rate of core $c_i \in C$, packet service time, etc.);

Find stochastic traffic models [such as (3) or (4)] and statistical parameters (e.g., mean and variance);

Such that the resulting models describe the asymptotic properties (e.g., steady-state behavior) of the network accurately and facilitate analysis [i.e., evaluation of cost functions defined in (2)] and optimization.

For example, starting from real multimedia traces, one can build an analytical model that captures the long-range dependencies as shown in (4). Then, using such a model, various performance and cost metrics such as packet loss probability and buffer size can be optimized.

3) Proposed Approaches: Existing research in NoC design has used the aforementioned methodology in deriving various traffic models. For instance, the authors in [167] introduce an analytical model based on identifying self-similar effects in multimedia traffic. These effects have important consequences for the design of on-chip multimedia systems since self-similar processes have properties which are completely different from traditional short-range dependent or Markovian processes that have been traditionally used in system-level analyses. Later, the authors in [156] empirically derived a comprehensive traffic model for NoCs which exposes both spatial and temporal dimensions of traffic via three statistical parameters: hop count, burstiness, and packet injection distribution. Interestingly enough, it has been recently reported that even though the traffic generated by programmable cores consists of multiple program phases, it can be used for on-chip traffic generation [147].

Unfortunately, the research in this area is still lagging due to the lack of well-defined NoC benchmarks. This situation has two primary reasons. First, the applications suitable for NoC platforms are typically very complex. For instance, it is common for applications to be partitioned among tens of processes (or more) in order to allow for evaluations of scheduling, mapping, etc. For general-purpose chip multiprocessors (CMPs), benchmarks such as SPLASH, originally designed for shared-memory multiprocessors, may be used. However, it is unclear if these benchmarks stress the NoCs effectively. Second, compared to traditional research areas like physical design where the design constraints are static (e.g., the aspect ratio of the blocks, number of wires between different blocks, etc.), NoC research requires detailed information about the dynamic behavior of the system; this is hard to obtain even using simulation or prototyping.

As a result, most researchers and designers still rely on synthetic traffic patterns such as uniform random and bit permutation traffic, to stress test a network design [39], [90]. While not entirely realistic, synthetic traffic models can be used to quickly obtain various performance metrics via rigorous analysis, as further detailed in Section VI.

4) Open Problems: A first step toward a unified approach for embedded platforms has been made recently via the open core protocol–IP benchmarking initiative [56]. Similarly, there have been initial steps toward releasing parallel benchmarks targeting the future CMPs [22]. Such initiatives can certainly boost the research progress in this area. However, there needs to be more research aimed at developing accurate traffic models, as well as in-depth studies that project NoC traffic for emerging workloads.

B. P2 Application Mapping

1) Motivation: At this stage, given a particular topology of the network, the application is typically described as a set of concurrent tasks that have been already assigned and scheduled onto a list of selected IP cores. The mapping problem for NoCs is to decide how to topologically place the selected set of cores onto the PEs of the network such that the metrics of interest are optimized. We note that "PE" simply means a placeholder connected to one of the network routers. In other words, mapping determines which IP core connects to which router in the network; this, obviously, greatly impacts both the performance and energy consumption of the NoC.

2) Problem Description:

Given an APCG $G(C, A)$ and a network topology $T(R, Ch)$;
Find a mapping function $\Omega : C \rightarrow R$ which maps each core $c_i \in C$ in the APCG to a router $r \in R$ in $T(R, Ch)$;
Such that the objective function $O(\text{Arch}, G)$ is optimized, subject to the constraints specified by $Const(\text{Arch}, G)$.

In this formulation, $O(\text{Arch}, G)$ and $Const(\text{Arch}, G)$ can be subsets of the metrics listed in (1) and (2). As an example, the objective can be minimizing the communication energy consumption, i.e.,

$$\min \left\{ Energy = \sum_{\forall a_{i,j}} v(a_{i,j}) \times E_{\text{bit}} \left(\Omega(c_i), \Omega(c_j) \right) \right\} \quad (5)$$

such that the bandwidth constraints for each link are satisfied. Here, $E_{\text{bit}}(\Omega(c_i), \Omega(c_j))$ is the energy required to send 1 b of data from the node where core c_i is mapped (i.e., $\Omega(c_i)$) to the node where core c_j is mapped; the remaining variables are defined in Definitions 1 and 2.

As pointed out in [60], the aforementioned mapping problem is a special case of the quadratic assignment problem where the goal is to minimize the sum of the products between a cost function and weights [e.g., see (5)] [129]. Depending on the cost, constraints, and flexibility allowed in the design, the mapping problem may have different forms and solution complexities. Indeed, the objective function and the constraints in (5) can be replaced with other performance or cost metrics in (1) and (2), respectively.

3) Proposed Approaches: The mapping problem for NoCs has been first addressed by the authors of [67], where a branch and bound algorithm is proposed to map a given set of IP cores onto a regular NoC architecture such that the total communication energy is minimized. At the same time, the performance of the resulting communication system is guaranteed to satisfy the specified design constraints through bandwidth reservation. Murali and De Micheli in [115] propose a mapping algorithm for NoC architectures which supports traffic splitting. In [158], the authors present a mapping algorithm to minimize the communication energy subject to bandwidth and latency constraints. A multi-objective mapping algorithm for mesh-based NoC architectures is presented in [7]. The proposed approach finds the Pareto mappings that optimize performance and power consumption using evolutionary computing techniques. Improving upon these studies, the authors in [60] propose a more general unified approach for application mapping and routing-path selection which considers both best effort and guaranteed service traffic.

We note that many mapping algorithms use (directly or indirectly) the average packet hop count as a cost function by relating the average number of packet hops to the communication energy consumption [70] or communication cost [115]. However, when PEs have different sizes, the communication latency and power consumption per unit of data exchanged between any two neighboring routers may differ significantly.

Therefore, as proposed in [114], embedding floorplanning information within the mapping loop becomes necessary to get more accurate energy/latency estimates.

In [113], the authors propose a methodology that exploits the heterogeneity of various models of operation (also called *use cases*) when mapping applications onto reconfigurable NoCs. This approach allocates the NoC resources based on different communication requirements (i.e., bandwidth and latency) and traffic patterns that characterize various use cases.

With ever increasing power density and cooling costs, it is important to reduce or eliminate the potential hot spots and have a thermally balanced design. In [72], a genetic algorithm is proposed to produce a thermally balanced design while minimizing the communication cost via placement.

4) Open Problems: One key component needed to solve the application mapping problem is the analytical model used for solution evaluation. For instance, if the goal is communication energy minimization, an accurate energy model is crucial. As aforementioned, a dynamic energy model based on average packet hop count has been used for 2-D mesh architectures. Similar analytical models are needed when the underlying architecture changes (e.g., mapping for irregular architectures) or the optimization objective changes (e.g., mapping to maximize performance rather than energy).

Along the same lines, effective performance models (such as those discussed in Section VI) are needed, because the choice of application mapping heavily impacts the communication performance of NoCs. In order to help optimization, such analytical models should distinguish, given any two intermediate mapping solutions S_i and S_j, whether S_i is better than S_j or vice versa. However, as opposed to energy models, deriving a good performance model is more difficult. The idea of using the average packet hop count is applicable only if the network is not congested. However, in most on-chip applications, the queuing delay of a packet (or flit) dominates; therefore, estimation based on average packet hop may not be good enough.

Finally, efficient techniques for *run-time* mapping and management of applications are needed. Toward this end, software development and code placement for embedded multiprocessors are discussed in [52]. Similarly, for applications launched dynamically, run-time mechanisms for mapping [3], [35] and/or migrating [18] are needed. Since execution time and arrival order of applications are not known *a priori*, finding optimal solutions is difficult and remains a big challenge.

C. P3 Application Scheduling

1) Motivation: Another important problem in NoC design is communication and task scheduling. Although scheduling is a traditional topic in computer science, most previous work focuses on maximizing performance [137], [176]. More recently, energy-aware scheduling techniques for hard real-time [57], [143], [152] and distributed [101], [109] systems have also been introduced, but they address only the bus-based or P2P communication. Without taking into consideration the complex effects of the network (e.g., congestion) which may change dynamically during task execution, such techniques cannot be directly applied to NoC scheduling.

2) Problem Description: First, we give some useful definitions and then formulate the problem of energy-aware scheduling for heterogeneous NoCs under deterministic routing. Unlike Definition 1 in Section II (which describes the application at core level), when dealing with scheduling, the target application is typically described at *task/transaction* level. For instance, the IP cores shown as dotted boxes in Fig. 1 need to be removed for scheduling purposes, since tasks are not bound yet to any IP core in the library. Hence, we need a new set of definitions.

Definition 3: A *communication task graph* (CTG) $G' = G'(S_T, A_T)$ is a directed acyclic graph, where each vertex represents a computational module of the application referred to as a task $t_i \in S_T$. Each t_i is annotated with relevant information such as the execution time on each type of PE, energy consumption (e^i_j) when executed on the jth PE, task deadlines $(d(t_i))$, etc. Each directed arc $a_{i,j} \in A_T$ characterizes the communication (or control) dependence between tasks t_i and t_j. Each $a_{i,j}$ has an associated label $v(a_{i,j})$, which stands for the communication volume (in bits) exchanged between tasks t_i and t_j.

Since both communication transactions and task execution need to be scheduled, Definition 2 in Section II also needs to be extended to include the behavior of computational nodes.

Definition 4: A NoC *architecture characterization graph* (ACG) can be uniquely described by the four-tuple $\text{Arch}(S_P, T(R, Ch), P_R, \Omega(S_P))$, where $T(R, Ch)$, P_R, and $\Omega(S_P)$ are defined in Definition 2, while S_P represents the set of PEs in the system as in [68].

Using these definitions, the energy-aware scheduling problem for heterogeneous NoC architectures under real-time constraints can be described as follows:

Given a CTG and an ACG;

Find a mapping function $M()$ from the set of tasks (S_T) to the set of PEs (S_P), together with a start time for each task and communication transaction;

Such that the objective function $O(\text{Arch}, G)$ is optimized subject to the constraints specified by $\text{Const}(\text{Arch}, G)$.

Again, $O(\text{Arch}, G)$ and $\text{Const}(\text{Arch}, G)$ consist of a subset of the metrics listed in (1) and (2). For example, the energy-aware scheduling can be formulated as follows:

$$\min \left\{ Energy = \sum_{\forall t_i} e^i_{M(t_i)} + \sum_{\forall a_{i,j}} v(a_{i,j}) \times e\left(P_{R_{M(t_i),M(t_j)}}\right) \right\} \quad (6)$$

such that

1) All the control/data dependencies are satisfied.
2) All the tasks t_i, for which deadlines $d(t_i)$ are specified, finish their execution before or at their respective deadlines, where $e^i_{M(t_i)}$ represents the computation energy on PE i where task t_i is mapped and $e(P_{R_{M(t_i),M(t_j)}})$ represents the communication energy spent to send 1 b of information from PE $M(t_i)$ to PE $M(t_j)$ under the routing policy P_R.

3) Proposed Approaches: An algorithm to solve the aforementioned problem was proposed in [68]. The algorithm is based on list scheduling; it first allocates more slack to those tasks whose mapping onto PEs has a larger impact on energy consumption and performance of the application. This approach

minimizes the overall energy consumption of the system, while guaranteeing the hard deadlines imposed on tasks. This is different, for instance, from the integer-linear-program-based approach in [169] which minimizes only the interprocessor communication and cannot guarantee hard deadlines. The work in [81] presents a scheduling and arbitration policy for NoC that uses code division multiple access techniques. These techniques provide higher interconnect bandwidth by modulating packets via orthogonal codes.

In [138], the authors consider dynamic applications running on a multiprocessor NoC as a set of independent jobs and propose exact timing models that effectively model both computation and communication of a job. Similarly, the work in [161] extends the NoC scheduling to consider multiple communication patterns (or scenarios). By allowing the bandwidth to be shared among multiple communication scenarios, a better resource utilization of the NoC is obtained.

It should also be noted that dynamic voltage scaling (DVS) can be used in conjunction with scheduling in order to minimize the overall energy consumption. Such techniques have been proposed in the past for both bus- [57], [143] and NoC-based communications [151], [169]. In these approaches, voltage scaling is applied to tasks and/or communication to minimize the power consumption, while accounting for the DVS overhead and satisfying the application deadlines.

4) Open Problems: Although we discuss the mapping and scheduling problems separately, they can also represent a joint optimization problem. However, since they are both very hard problems to solve, such an integrated approach remains an open problem. This is particularly challenging for NoCs since communication delay is difficult to estimate, and therefore, deriving accurate models that can be used to guarantee hard deadlines is a huge problem by itself.

The algorithm in [68] targets real-time or DSP applications where the worst case task-execution time and inter-task communication volume are precharacterized. Such a model allows partitioning and scheduling to be done at compile time. However, this kind of algorithm cannot be directly applied to applications with conditional branches or applications that enter and leave the system dynamically. For applications whose behavior cannot be precisely predicted at compile time, *online* scheduling approaches are needed. Consequently, algorithms to support such general scenarios, with static or dynamic priorities assigned to tasks, also remain open research problems.

IV. Communication Paradigm

Knowing the application mapping and traffic characteristics helps with various decisions concerning the communication paradigm. Indeed, the effectiveness of the NoC communication infrastructure depends on the routing policy and switching protocol, quality of service (QoS) and congestion control, energy and power management, techniques for increased reliability, and fault tolerance.

A. P4 Packet Routing

1) Motivation: Given an underlying topology, the routing protocol determines the actual route taken by a message. The routing protocol is important as it impacts all network metrics, namely, latency (as the hop count is directly affected by the actual route), throughput (as congestion depends on the ability of the routing protocol to load balance), power dissipation (as each hop incurs a router energy overhead), QoS (as routing can be used to channel different message flows along distinct paths to avoid interference), and finally, reliability (as the routing protocol needs to choose routes that avoid faults).

2) Problem Description:

Given an application graph APCG (or CTG) and a network topology $T(R, Ch)$;

Find the routing protocol $P_R(r, Src, Dst)$ that determines the outgoing channel at router $r \in R$ for all packets traveling from source router Src to destination router Dst;

Such that $O(\text{Arch}, G)$ is optimized subject to $Const(\text{Arch}, G)$.

Next, we provide a concrete example of the objective function $O(\text{Arch}, G)$ and constraint function $\text{Const}(\text{Arch}, G)$, assuming that we want to minimize the average distance traveled by packets in the network, with a constraint on the maximum distance between any pair of nodes. In this case, the problem formulation becomes

$$\min \left\{ \sum_{\forall a_{i,j}} v(a_{i,j}) \times d_{i,j}\left(P_R(r, i, j)\right) \right\} \tag{7}$$

such that

$$\max_{a_{i,j}} d_{i,j}(P_R) \leq D_{\max} \tag{8}$$

where $v(a_{i,j})$ is the total volume of data sent from node i to j, $d_{i,j}$ is the distance between nodes i and j under the routing policy $P_R(r, Src, Dst)$, and $D_{\max}$ is the maximum allowable inter-node distance; the other variables are taken from Definitions 1 and 2. Intuitively, (7) minimizes the weighted sum of inter-node distances, where the weights are the inter-node communication volumes. Equation (8) sets the limit for the maximum distance between any pair of nodes in the network.

3) Proposed Approaches: Routing has been extensively studied in multichassis interconnection networks, many of which have been leveraged in on-chip networks. One example is the dimension-ordered routing which routes packets in one dimension, then moves on to the next dimension, until the final destination is reached. While such a technique is very popular due to its simplicity, adaptive routing techniques (e.g., turn model routing and planar adaptive routing [39], [45]) can provide better throughput and fault tolerance by allowing alternative paths, depending on the network congestion and runtime faults. Oblivious routing algorithms which generate routes without any knowledge of traffic have also been extensively studied in the context of multichassis interconnection networks [144] and can be relevant to on-chip networks due to their low overhead [165].

New routing protocols have also been investigated specifically for NoCs. For instance, *deflective routing* in [118] routes packets to one of the free output channels belonging to a minimal path; if this is not possible, then packets are misrouted. Techniques have been also proposed to dynamically switch between deterministic and adaptive routing [69].

The application-specific customization of routing protocols (e.g., [70] and [115]) and techniques to provide low overhead routing algorithms with high path diversity [1], [111] have been explored. With NoCs being increasingly concerned with

power, thermal, and reliability issues, there exists recent work proposing thermal- [146] and reliability-aware [105] routing algorithms. These directions are discussed later in P7 and P8.

4) Open Problems: While some existing research into routing algorithms for off-chip interconnection networks can be leveraged for NoCs, the significantly different constraints of on-chip implementations lead to new challenges. First, the ultralow latencies [54] and very high frequencies [166] of some NoCs make it difficult to incorporate sophisticated routing algorithms such as adaptive routing. The tight power constraints and reliability issues also lead to challenges in power-aware and fault-tolerant routing. With static topology irregularity, it is difficult to find minimal routes that can avoid deadlock and livelock situations (e.g., [29]). This is a research direction that needs more attention in the future. Relying on dimension-ordered routing as the escape routing function in irregular topologies becomes difficult, and implementations typically require tables that incur delay, area, and power overheads [25]. To date, the vast majority of NoC routing solutions have focused on unicasting, i.e., sending from one PE to another. Support for on-chip multicast [49] needs to be considered also, with emphasis on very lightweight solutions so that tight on-chip constraints can be met.

B. P5 Switching Technique

1) Motivation: Switching, also called flow control,[2] governs the way in which messages are forwarded through the network. Typically, the messages are broken down into *flow control units* (*flits*) which represent the smallest unit of flow control. The switching algorithm then determines *if* and *when* flits should be buffered, forwarded, or simply dropped [39], [45]. As a result, the switching algorithm has the most direct impact on router microarchitecture and pipeline.

2) Problem Description:

Given a routing policy $P_R(r, Src, Dst)$ at router r connecting any two channels ch_i, $ch_j \in Ch$ along the route for a flit arriving from ch_i;

Find a protocol to multiplex flits onto ch_j via forwarding the flits through the router r;

Such that ch_j is maximally utilized, yet the overhead of router r is low.

3) Proposed Approaches: Among the commonly used switching techniques in interconnection networks, wormhole switching seems the most promising for NoCs due to the limited availability of buffering resources and tight latency requirements. Virtual channels (VCs) [41] are widely used in off-chip interconnection networks and are naturally adopted for NoC design to improve network bandwidth and tackle deadlock. However, as the design requirements change dramatically, the underlying substrate presents new opportunities for designing flow control algorithms.

Early work on NoC flow control aggressively drives down the router delay to a single cycle, through static compiler scheduling of network switching operations [164], dedicated look-ahead signals for setting up the switch ahead of time [54], by speculatively allocating resources to move the latency associated with resource allocation and multiplexing off the critical path at low traffic loads [110], [133], or through advanced reservation of resources [132]. While most studies focus on packet switching, several papers investigate the potential of circuit switching, as a way to obviate the arbitration and buffering overheads of packet-switched routers [48], [60], [173], [174].

Several techniques tackle NoC throughput, such as dynamically varying the number of VCs assigned to each port, to better adapt to the traffic load [116]. Express VCs aggressively drive down the router latency to just link latency while extending throughput by having VCs that are statically defined to cross multiple hops [88]. Tackling latency and throughput simultaneously, layered switching [100] hybridizes wormhole and cut-through switching by allocating groups of data words which are larger than flits but smaller than packets.

4) Open Problems: There is still a significant latency/throughput gap between the state-of-the-art NoCs and the ideal interconnect fabric of dedicated wires [88]. This disparity largely lies in the complex routers necessary at each hop for delivering ultralow latency and/or high bandwidth. In order for NoCs to be an efficient and effective replacement of dedicated wires as the primary communication fabric, there is a need for new switching techniques that can obviate this router overhead and truly deliver the energy–delay–throughput of dedicated wires.

C. P6 QoS and Congestion Control

1) Motivation: Conventional packet-switched NoCs multiplex message flows on links and share resources among these flows. While this results in high throughput, it also leads to unpredictable delays per individual message flows. For many applications with real-time deadlines, this nondeterminism can substantially degrade the overall application performance. Thus, there is a need for research into NoCs that can provide deterministic bounds for communication delay and throughput.

2) Problem Description:

Given a NoC architecture Arch$(T(R, Ch), P_R, \Omega(C))$ and an application graph APCG (or CTG);

Find a resource allocation strategy (e.g., size of output buffers of router $r \in R$, bandwidth of channel $ch_i \in Ch$, etc.) and/or packet injection rates in the network;

Such that no user-level transaction experiences an end-to-end network performance greater than a specific threshold $\{L_{\max}, B_{\max}\}$, where $L_{\max}$ is a specified network latency and $B_{\max}$ is a specified bandwidth. These thresholds can be further specified as a function of different transaction classes, e.g., $L_{\max,\mathrm{GT}}$ and $L_{\max,\mathrm{BE}}$, where GT represents guaranteed throughput and BE represents best effort traffic.

3) Proposed Approaches: QoS in NoCs is typically handled through three types of approaches. First, resources such as VCs can be prereserved with a fair mechanism for allocating resources between different traffic flows [20], [53], [95], [97], [108]. Second, multiple priority levels can be supported within the network such that the urgent traffic can have a higher priority over the regular traffic [13], [24], [61], [106]. Techniques to ensure global fairness to network hot spots have been proposed in [92]. Finally, QoS-aware congestion control algorithms have been proposed to avoid the spikes in delay when the traffic load

[2]The two terms "switching" and "flow control" have been used interchangeably in leading NoC texts [39], [45], [107].

approaches saturation by having congestion control at the NI regulate traffic and to ensure fairness [26], [46], [118], [122].

4) Open Problems: Future CMPs and MPSoCs impose increasingly more QoS demands on NoCs; yet, support for QoS has to be extremely lightweight. Cache-coherent CMPs would benefit from NoC being able to preserve the ordering semantics of a bus, thereby easing consistency support. Being able to provide guarantees on packet deliveries such as snoop responses will also ease protocol design and lower the protocol overhead. Both SoCs and CMPs will benefit from NoCs that can support dynamically defined QoS levels and needs, as well as dynamically defined partitions of the NoC that support different QoS.

Furthermore, it will greatly strengthen the fundamental basis of NoCs to have research into analytical models that can estimate network latency and/or throughput for arbitrary traffic patterns, as these can be used to feed into QoS engines with low implementation overhead.

D. P7 Power and Thermal Management

1) Motivation: Due to concerns on battery lifetime, cooling, and thermal budgets, power issues are at the forefront of NoC design. Indeed, several NoC prototypes show NoCs taking a substantial portion of system power, e.g., $\sim$40% in the Massachusetts Institute of Technology RAW chip [163] and $\sim$30% in the Intel 80-core teraflop chip [166].

2) Problem Description: We denote by Act the set of actions that a power manager in NoC can take in response to application requests for service in order to tackle the power and temperature problems. This set can include changing the voltage and/or frequency of a set of cores, scaling the voltages of NoC links, etc., in response to workload variation. The goal is as follows:

Given an application graph APCG (or CTG) and its service requests, a network topology $T(R, Ch)$, the set of actions Act the system can take on the state space Σ, and the power-related parameters for the NoC and cores such as various operation modes with different power/performance costs;

Find an optimal policy for dynamic management on the state space Σ and the actions specified in Act;

Such that the metrics of interest (e.g., average or peak power consumption, energy consumption, and average or peak temperature) are minimized (or constrained) with respect to a set of given performance constraints.

From this description, it becomes clear that the power manager has to solve an optimization problem under performance constraints. Obviously, more accurate tradeoffs can be made based on more complex decision models. This has to do with the amount of "history" that the power manager is able to handle, the type of policies suited for the problem at hand (deterministic versus randomized), etc. The important issue to note is that while various approaches do exist in the literature, the power management problem becomes much more complicated for NoC-based systems since, in this case, the state of the system and manager actions have to take into consideration the behavior of the entire network.

3) Proposed Approaches: Seminal work on router power modeling [131] along with position papers that highlight the importance of NoC power consumption [15], [40], [74] motivates research into low-power NoCs. There has been research into run-time NoC power management using DVS on links [145], as well as shutting links down based on their actual utilization [80], [155]. Globally asynchronous locally synchronous (GALS) approaches to dynamic voltage and frequency scaling further leverage the existing boundaries between various clocking domains [14], [124]. For instance, the work in [124], uses feedback control to manage the voltage and frequency of various islands in the network in order to save energy. Aside from average power, peak power control mechanisms for NoCs have also been explored [19].

Power has also been investigated in tandem with other metrics, with [154] achieving power management in the presence of QoS constraints. Power savings are achieved together with improved reliability through error coding schemes in [17] and [175].

Thermal dissipation is another metric of interest, and mechanisms have been investigated to control peak power with respect to its impact on thermal dissipation [72], [146]. The work presented in [38] explores thermally aware task scheduling for MPSoCs. The authors in [162] propose a 3-D MPSoC thermal optimization algorithm that involves task assignment, scheduling, and voltage scaling.

4) Open Problems: With NoCs facing highly constrained power envelopes, run-time power management techniques have to be liberally used to reduce peak power consumption so as to avoid thermal emergencies. Challenges remain in practical routing algorithms in the presence of extensive router and link power management. Such exploration should perhaps be done in conjunction with circuit- and/or OS-related research.

Another wide-open area for research is related to *distributed* control strategies for power and thermal management in NoCs. Techniques like those in [124] and [154] are based on a centralized power manager but perhaps, relying only on localized information, may have its own advantages for NoCs; whether this is indeed the case remains to be investigated. It is important to note, again, that the accuracy of the energy models is crucial for these optimization techniques. Ideally, such models should target both dynamic and static power dissipations. While there exist preliminary efforts in this direction (e.g., [168] where adaptive body biasing is used to minimize static energy consumption), more work is needed to achieve practical solutions.

E. P8 Reliability and Fault Tolerance

1) Motivation: As CMOS technology approaches the nanoscale domain, there is an increasing need for studying how NoC architectures can tolerate faults and underlying process variations. For instance, shrinking transistor sizes, smaller interconnect features, and higher operating frequencies of current CMOS circuits lead to higher soft-error rates and an increasing number of timing violations [150]. Moreover, the combination of smaller devices and voltage scaling in future technologies will likely result in increased susceptibility to transient faults. Therefore, in order to reduce the cost of design and verification, the future SoC architectures need to rely on fault-tolerant approaches.

2) Problem Description: Two types of faults need to be addressed in future NoC architectures: hard (or permanent) and soft (or transient) faults. Permanent faults represent irreversible

physical changes in the structure of logic devices (e.g., due to electromigration). Transient faults are caused by particle strikes (e.g., neutron and alpha particles), power supply and interconnect noise, electromagnetic interference, or electrostatic discharge. The rate of occurrence of transient faults increases with reducing the feature size of logic devices; therefore, accurate fault models are critical for addressing various reliability and fault-tolerance issues. For instance, fault models may characterize permanent and transient faults in routers and links, and packet probability of being corrupted by data upsets or simply dropped due to buffers overflow. Also, synchronization failures at clock interfaces should be considered when developing a fault model for multiple clock domain designs.

Given an application graph APCG (or CTG), a network topology $T(R, Ch)$, and a fault model that accurately captures the probability of permanent node/link failures, data upsets at nodes and links, packet losses (e.g., due to buffer overflow, corruption of header information, etc.), and synchronization errors;

Find a set of mechanisms for efficient and reliable data delivery (e.g., set of minimal alternate paths, minimum number of retransmissions, deadlock-free routing strategy, etc.);

Such that packets are delivered to their destination with minimum impact on performance and other cost metrics, despite the transient and permanent failures in the network.

In order to deal with both transient and permanent faults, the routing algorithms should exploit the path diversity and be capable of generating multiple routes for every possible $Src{-}Dst$ pair in the network. If an approach based on acknowledgment request protocol is used to mitigate transient errors, then the problem description should be modified to consider the optimal number of retransmissions.

3) Proposed Approaches: Fault-tolerant multichip interconnection networks have been investigated, mostly in the areas of fault-tolerant routing or microarchitecture [45]. For NoCs, a fault-tolerant communication approach was proposed in [23]. This approach is based on probabilistic broadcast where packets are forwarded randomly to the neighboring nodes. A theoretical model explaining the stochastic communication and relating the node coverage to the underlying properties of a grid topology is also provided. Similarly, the studies in [135] and [139] explore how NoC routing algorithms can route around faults and sustain network functionality in the midst of faults.

Researchers have also modeled the interaction between various NoC metrics, like delay, throughput, power, and reliability [50]. Several studies look specifically into router design and ways to improve the NoC reliability through microarchitectural innovations that go beyond the expensive alternative of having redundant hardware [6].

In [140], the authors consider buffered flow control techniques with different area and power tradeoffs and evaluate their fault-tolerance overhead. Similarly, power consumption at link-level and end-to-end data protection in NoCs is analyzed in [75]. In [17] and [112], the authors explore energy efficiency and reliability of error correction at the receiver side and present techniques for combined energy minimization and reliability optimization.

4) Open Problems: With devices moving into deep submicrometer technologies, reliability becomes a very important issue. However, research into NoC reliability is still in its infancy, and thus, realistic fault models that are a good representation of physical realities for NoCs are needed. Research exploring the trends in soft-error rates for combinational logic (e.g., [153]) can potentially be relevant to router microarchitectures as well. Future work that will critically impact the NoC power–performance–reliability tradeoff includes the cost effectiveness of providing fault tolerance while maintaining suitable levels of fault isolation and containment. With techniques exploring fault detection at the router level (i.e., packet/flit drops), correction and recovery will be essential in future NoCs, but again, the cost and impact on power and performance must be low.

V. Communication Infrastructure

High-level decisions made about the NoC design must be translated into lower level design layers. This section examines the topology, router, channel, and various clocking strategies that can be used for NoC design.

A. P9 Topology Design

1) Motivation: The ability of the network to efficiently disseminate information depends largely on the underlying topology. Indeed, besides having a paramount effect on the network bandwidth, latency, throughput, overall area, fault tolerance, and power consumption, the topology plays an important role in designing the routing strategy (see P4) and mapping the IP cores to the network (see P2).

2) Problem Description:

Given an application graph APCG (or CTG);

Find network topology $T(R, Ch)$ (i.e., minimum number of routers, PEs, links, and graph structure interconnecting them);

Such that the objective function $O(\text{Arch}, G)$ is optimized subject to the constraints specified by $\text{Const}(\text{Arch}, G)$.

In this formulation, $O(\text{Arch}, G)$ can be one of the metrics listed in (1). For example, (7) can be used as an objective function, since the network topology directly affects the internode distances $d_{i,j}$. Likewise, the constraints $\text{Const}(\text{Arch}, G)$ correspond to (2) (e.g., power budget and wiring resources).

3) Proposed Approaches: The simplicity and regularity of grid structures make design approaches based on such a modular topologies very attractive. More precisely, regularity improves timing closure, reduces dependence on interconnect scalability, and enables the use of high-performance circuits. Typically, 1-D (e.g., ring [136]) and 2-D topologies (e.g., mesh and torus [54], [163]) are the default choices for NoC designers. Node clustering to obtain topologies like the concentrated mesh [9] and hierarchical star [93] is a viable alternative to amortize the router overhead and reduce latency. Higher radix networks like the flattened butterfly [86] reduce power and latency by reducing the number of intermediate routers and the wiring complexity over a conventional butterfly but they increase the number of long wires. In [171], the authors explore the energy consumption of various standard NoC topologies for different technology nodes. The work in [16] chooses the NoC topology from a given topology library for various power/performance and area tradeoffs.

Despite the benefits of regular network topologies, customization is also desirable for several reasons. First, when the

size or shape of the cores varies widely, using regular topologies may waste area. Moreover, for real applications, the communication requirements of the components can vary widely. Designing the network to meet the requirements of highly communicating cores results in the under-utilization of other components, while designing it for the average case results in performance bottlenecks. Finally, for application-specific NoCs, a detailed *a priori* understanding of the communication workload can be exploited to fully customize the network topology [63], [119]. For instance, the approach proposed in [134] enables the automatic design of the communication architecture of a complex system using a library of predefined IP components. Similarly, the work in [159] presents a mixed-integer-linear-programming-based technique for NoC topology synthesis with the objective of minimizing the power consumption subject to performance constraints.

Interestingly enough, the two extreme points in the design space (i.e., completely regular and fully customized topologies) are not the only possible solutions for on-chip communication. Indeed, by adding a few long-range links, the performance of regular topologies can be significantly improved with minimal impact on area and energy consumption [121]. Recently, the idea of inserting long-range links into NoCs has been extended for on-chip radio-frequency links [32].

4) Open Problems: Generally speaking, the problem of optimal topology synthesis for a given application does not have a known theoretical solution. Although the synthesis of customized architectures is desirable, distorting the regular grid structure leads to various implementation issues such as complex floorplanning, uneven wire lengths, etc. Although we have discussed only planar substrates, 3-D die stacking provides the opportunity for higher radix topologies through the use of inter-die connections [85], [162], [177] and warrants further exploration.

B. P10 Router Design

1) Motivation: The design of a router involves determining the flow control techniques, number of VCs, buffer organization, switch design, pipelining strategy while adhering to target clock frequency and power budgets. All these issues require careful design since they have significant impact in terms of performance, power consumption, and area.

2) Problem Description: Obviously, the router design problem encompasses the implementation of solutions to problems P4–P8. It is difficult and perhaps of limited use to combine all these aspects into a single problem description. Since we focus on algorithmic aspects of NoC research, here, we give an example problem description for the buffer-sizing problem which has significant impact on area and performance figures of the design.

Given an application graph APCG (or CTG) and a network topology $T(R, Ch)$, the probability distribution of packet destinations, packet size, etc., and architecture-specific parameters such as the routing protocol $P_R(r, Src, Dst)$ and total available buffering space B;

Find the buffer size $l(ch, r)$ for *each* channel ch, at each router r in the network;

Such that the objective function $O(\text{Arch}, G)$ is optimized subject to the constraints specified by $\text{Const}(\text{Arch}, G)$.

For example, the objective function can minimize the average packet latency with a bound on the total buffer space, i.e.,

$$\min(\text{average packet latency}) \tag{9}$$

$$\text{such that } \sum_{\forall r} \sum_{\forall ch} l(ch, r) < B. \tag{10}$$

Similar to the previous formulations, other performance and cost metrics defined in (1) and (2) (e.g., power consumption and throughput) can replace the objective function and the constraints in this formulation.

The router microarchitecture design and VC planning can be formulated in a similar way. For instance, given the network topology and bandwidth requirements, one can design the router microarchitecture in terms of the number of pipeline stages, choice of allocation algorithms, speculation, and bypass techniques such that the power consumption and average packet latency through the router is minimized.

3) Proposed Approaches: The main focus in designing a router is to minimize the latency through it while meeting bandwidth requirements. Reservation [132] and speculation [110], [133] can be used to hide the routing and arbitration latencies and achieve a single-stage router design. Decoupled parallel arbiters and smaller crossbars for row and column connections can reduce contention probability and latency [84]. Moreover, techniques such as segmented crossbars, cut-through crossbars, and write-through buffers can be used to design low-power routers [170].

The impact of the number of VCs on performance varies with the network load. A lightly loaded network does not need many VCs, whereas a heavily loaded network does. A VC regulator which dynamically allocates VCs and buffers according to traffic conditions, thereby reducing total buffering requirements and saving area and power, is presented in [116]. Arbitration during VC allocation is another area of potential optimization. Free VC queues at each output port can effectively remove the need for VC arbitration by predetermining the order of grants [110].

An efficient algorithm for the buffer size allocation problem is proposed in [71]. The authors derive the blocking rate of each individual channel and then add more buffering resources only to the highly utilized channels. Similarly, the properties of on-chip buffers and gate-level area estimates are studied in [141]. Finally, advanced circuit-level techniques have been employed to achieve high-speed and low-power operation. For instance, the router presented in [166] employs a double-pumped pipeline stage to interleave alternate data bits using dual edge-triggered flip-flops; this optimization reduces the crossbar area by 50%. Similarly, serial on-chip links, partial crossbar activation, and low energy transmission coding techniques are used in the router design presented in [93].

4) Open Problems: Tools that enable microarchitecture exploration to tradeoff the latency and bandwidth of the router against power consumption can help NoC designers make the right design decisions for particular application requirements. Accurate performance analysis of on-chip routers under arbitrary input traffic and methodologies for choosing the correct design parameters such as optimal channel width, buffer depth, pipeline depth, and number of VCs for high performance and low power remain open problems. Finally, energy-efficient routers that can interface with a variety of IP cores designed

for legacy communication protocols with minimal performance overhead is an important challenge.

C. P11 Network Channel Design

1) Motivation: The links interconnecting the network routers also need to be designed efficiently in order to consume low power and area. The ideal interconnect should be such that its performance and cost come close to that of just the network channels (or links), with the performance delivered and power consumed by the network channel largely determined by the signaling techniques.

The width of the network channel is important in determining the latency through the channel and area cost. In the following, we assume that $w(ch) = W$. Then, the bandwidth of a network channel is given by

$$BW = f_{\text{ch}} \times W \tag{11}$$

where f_{ch} is the channel operating frequency. Increasing W reduces the contention-free message latency, which is given by

$$L_0 = H \times (t_r + t_s + t_L) + \max(t_s, t_L) \left\lceil \frac{L}{W} \right\rceil \tag{12}$$

for a message consisting of L bits (assuming wormhole routing, see P5), where H represents the hop count, t_r, t_s, and t_L are the latencies associated with making the routing decision and traversing the router and the link, respectively. While these relations suggest possible benefits from increasing W, doing so actually increases the use of wiring resources. Moreover, wire pitch and spacing affect interference between the wires. A larger W also implies wider input buffers in the routers which can also result in a larger buffer area (see Section II). Finally, wider channels result in increased switching resources in the crossbar at each node, both in terms of area and power, usually as a quadratic function of the number of links.

2) Problem Description:

Given an application graph APCG (or CTG) and the network topology $T(R, Ch)$;

Find a signaling technique and the channel width W;

Such that the objective function $O(\text{Arch}, G)$ is optimized subject to the constraints specified by $\text{Const}(\text{Arch}, G)$.

For example, the objective can be minimizing the network latency (or maximizing the throughput) under power consumption and wiring constraints.

3) Proposed Approaches: Nonuniform channel capacity allocation is presented in [59] where the traffic is assumed to be heterogeneous and critical delay requirements vary significantly. In addition to the effects aforementioned, the choice of W has implications on the wire sizing and spacing, which affect the channel operating frequency. Hence, the bandwidth cannot be optimized by simply considering f_{ch} and W separately. Pileggi and Lin [98] discuss maximizing channel throughput by controlling the size and spacing of wires, as well as their number. In [78], the authors discuss a framework for equalized interconnect design for on-chip networks. The proposed approach finds the best link design for target throughput, power, and area constraints, and enables architectural optimization for energy-efficiency.

In [117], the authors present a delay-insensitive current-mode signaling technique for high-performance NoC links. Low-swing, pulse-mode, and differential signaling techniques [76] can be used to improve performance and reduce power.

4) Open Problems: Determining the optimal channel width may not be possible as it is directly dependent on the application and other factors involved in the network design. As such, tools for analyzing the tradeoffs involving the channel width subject to wiring and area constraints are needed for a fair comparison among different communication architectures.

Alternatives to wire channels present interesting opportunities for future research. For instance, analog–digital hybrid routing approaches [102] and RF interconnects have been considered as an alternative to traditional on-chip repeated interconnects [32], [181], but more work is needed to make such approaches applicable to real designs.

D. P12 Floorplanning and Layout Design

1) Motivation: Standard tile sizes help in controlling the link lengths and ensuring that link delays do not limit the operating frequency. However, if the size of the network tiles varies significantly or irregular topologies are used, the floorplanning step becomes mandatory. In this case, emphasis needs to be put on the shape and placement of tiles so as to control the link lengths. Reducing the total interconnect length is also important for reducing the power dissipation across the links. Another problem is the placement of special tiles like those connected to peripheral devices (e.g., memory controllers, and I/Os) so as to minimize the average latency to these devices. In addition to link length, the goal here is to minimize link area by routing links over logic or caches as much as possible.

2) Problem Description:

Given the NoC architecture $\text{Arch}(T(R, Ch), P_R, \Omega(C))$ and the sizes of the cores;

Find a floorplan (e.g., determine $\text{Pos}(r) \forall r \in R$) which minimizes the objective function $O(\text{Arch}, C, \Omega(C))$;

Subject to the constraints in $\text{Const}(A, C)$.

The total area and inter-router distances can be used as either objective function or constraints. For instance, this can be the Manhattan distance between different logic blocks. Additional constraints can be the maximum length for a given network channel, aspect ratios of the logic blocks, latency, and communication bandwidth requirements between blocks.

3) Proposed Approaches: In [127], the authors explore the layout, performance, and power tradeoffs in mesh-based NoCs. They identify two extremes: routing over logic and routing with dedicated channels. One approach places restrictions on the placement of repeaters and the reuse of IP cores, while the other uses more area but provides a better signal integrity on the links. The authors in [5] present a comparison in terms of performance, area, and power scalability between crossbar designs within a pre-existing communication fabric and the NoC approach at layout level. Although the area overhead of the NoC approach is higher, its performance and energy figures are better.

Floorplanning for regular topologies is addressed in [179] based on planarization process constrained by aspect ratios, preservation of regularity and hierarchy, and minimization of timing delay and wire power. Integrating the floorplanning with application mapping and topology synthesis driven by power constraints is discussed in [114] and [160]. The focus of

these studies is mainly on minimizing latency and link lengths through intelligent floorplanning. Although, in this section, we only consider cores of fixed size, Kumar *et al.* in [89] argue that this may not be necessarily the case in practice since, while designing NoCs, cores, and caches, area overhead in one component may force a reduction in the area of another.

4) Open Problems: Floorplanning for arbitrary topologies remains largely an open problem; therefore, tool support for exploring different floorplans under concurrent constraints of area, power, latency, bandwidth, regularity, and tile aspect ratios is needed. Different from general application-specific integrated circuits, floorplanning for NoCs may eventually consider the placement of routers and repeaters for latency insensitive operation [28]. Furthermore, the routability of the global network channel for maximum performance and well-controlled coupling effects needs to be taken into account.

E. P13 Clocking and Power Distribution

1) Motivation: The traditional approach of designing a fully synchronous chip with a single global clock rapidly runs out of steam due to smaller process geometries, larger wire delays, and higher levels of integration of multiple cores on large dies. The large effort required for skew control and the significant power consumption of the global clock call for alternative clocking strategies [12]. Indeed, in addition to multiple frequencies, different cores can have their own optimal supply voltage to allow for fine-grain power/performance management.

2) Problem Description:

Given the die size, network topology $T(R, Ch)$, set of cores, and their voltage and frequency requirements;

Find the clocking strategy and appropriate voltage domains;

Such that the average communication latency, power consumption, and design effort are minimized.

3) Proposed Approaches: Strategies such as asynchronous or mesochronous clocking [8], [148], [157] are alternatives that hold the promise of simplifying timing closure and global clock distribution. For instance, an approach to minimize the strict skew requirements without going fully asynchronous using all-in-phase clocking is presented in [149]. In [157], the authors present a mesochronous clocking strategy that avoids timing-related errors while maintaining a globally synchronous system perspective. The 80-tile teraflop NoC presented in [166] employs phase-tolerant mesochronous interfaces between the routers with first-in first-out (FIFO)-based synchronization. Similarly, latency insensitive or synchronous elastic systems are developed to exploit the inherent advantages of synchronous design while decoupling the functionality from the channel delays [28], [37].

The GALS approach has been used with several tile-based multiprocessor implementations [180]. However, there are extra costs in terms of synchronization latency and power that need to be considered. Chelcea and Nowick [33] discuss interfacing different clock domains which is essential for implementing globally asynchronous systems. A systematic comparison between asynchronous and GALS implementations of a NoC is presented in [148]. The authors conclude that, while the two approaches result in similar silicon area, power consumption, and bandwidth, the asynchronous implementation has a clear advantage in terms of average packet latency. In [27], the authors propose asynchronous delay-insensitive links to support the GALS NoC paradigm. This approach removes the constraints on wire propagation delays and enables designing links of any width with low wire and logic overhead. Finally, a design methodology for iteratively partitioning a NoC into multiple voltage–frequency islands to minimize energy consumption under performance constraints is proposed in [124]. This approach controls the utilization of interface queues across different clock domains based on a linear system model.

4) Open Problems: Open problems in this area include the robust design of clock crossing synchronizers with minimal latency penalty and low power consumption, since locally generated clocks for GALS SoCs are prone to synchronization failures due to clock delays [43]. Recent research in resonant clocking shows promise for reducing power and delivering high performance [31]; however, the use of resonant clocking with NoCs has yet to be investigated. Also, while controlling NoCs with multiple voltage–frequency islands has been discussed in [124], techniques that consider other control objectives such as chip temperature, power consumption, and nonlinear effects are needed.

VI. NoC Evaluation and Validation

After the decisions regarding the communication paradigm and infrastructure are made, an evaluation of the system is necessary to ensure its compliance with the initial specs. This can be a fast power/performance analysis step during optimization stages or simulation and prototyping as the design converges.

A. P14 Analysis and Simulation

1) Motivation: Fast and accurate approaches for analyzing critical metrics such as performance, power consumption, or system fault tolerance are important to guide the design process. However, in order to be used within an optimization loop or make early design choices, the analysis techniques need to be tractable and provide meaningful feedback to designers. At later design phases, one can obtain more accurate estimates through simulation.

2) Problem Description:

Given the NoC architecture Arch$(T(R, Ch), P_R, \Omega(C))$ and an application graph APCG (or CTG);

Find the *expected* (or average) values of the variables of interest (e.g., packet latency, power consumption, etc.) as a function of the architecture and application characteristics.

For instance, the NoC average packet latency for a packet moving from the source node src to the destination node dst is

$$L_{\text{src-dst}} = W_{\text{src}} + \sum_{i \in \text{Path(src,dst)}} (W_i + T_S) \qquad (13)$$

where W_{src} is the queuing delay at the source, T_S is the average service time (including the serialization latency), and W_i is the delay at the intermediate nodes on the path from src to dst [i.e., Path(src, dst)] due to queuing and blocking. Consequently, a performance analysis technique needs to find the steady-state behavior and provide estimates for *each* of these components.

Of course, finding accurate estimates for steady-state values is a complicated problem, and various mathematical formalisms available for analysis (e.g., queuing analysis, Markov chains,

etc.) are based on simplifying assumptions to make the problem tractable. Furthermore, the impact of the initial conditions on the simulated network should also be considered. Finally, in order to draw reliable conclusions, one needs to know the level of confidence of these simulation results and when the desired level of confidence has been reached.

3) Proposed Approaches: Communication latency and network bandwidth are common performance metrics of interest. While it is relatively easy to find the communication latency for guaranteed service traffic [42], [108], analyzing the average latency for best effort traffic is a challenging task. Therefore, the average hop count or free packet delay are commonly used to approximate the average packet latency [115], [159]. Regarding the best effort traffic, the authors in [123] present a mathematical model of on-chip routers which can be used to derive analytical expressions for the average packet latency and saturation throughput of a NoC under wormhole routing. Other techniques for analyzing the average communication latency in networks are proposed in [44] and [66]. While not directly applicable to NoC performance analysis, these approaches can be used as a starting point and then account for NoC-specific constraints, such as application-specific traffic and on-chip router parameters.

Analytical power models for early-stage power estimation in NoCs have also been investigated, starting with [131] which models the power consumed by multichip interconnection networks; this generated follow-up works that specifically target on-chip network power dissipation [10], [30], [34], [47], [82], [146].

Simulation-based approaches for the architectural exploration of on-chip networks are presented in [87], [126], and [172]. In [103], the authors present a NoC model that can be used with a multiprocessor real-time operating system to analyze the behavior of a complex system. The study in [128] presents a comparative evaluation of various NoC architectures with realistic traffic models in terms of latency, throughput, and energy. In [51], the authors present an emulation framework for exploring and comparing a wide range of NoC solutions.

4) Open Problems: Performance analysis largely depends on various simplifying assumptions on the network or traffic characteristics (e.g., uniform traffic versus bursty traffic) and typically assumes deterministic routing due to the difficulty in handling the more general problem. Approaches that relax the Markovian assumption and analytical power consumption models that accurately account for the application and architecture characteristics are highly needed.

The major issue with simulation-based approaches is the tradeoff between the level of implementation detail and simulation time [73]. Detailed models can deliver very accurate results, but the simulation time can be prohibitive. Realistic synthetic trace simulation [104], [167] or hardware acceleration [51] can be used for improving the simulation speed; therefore, these are wide-open directions for research.

B. P15 Prototyping, Testing, and Verification

1) Motivation: While simulation offers flexibility for power–performance evaluations under various network parameters, it still relies on many approximations that may affect the accuracy of the results. Prototyping can be further used to improve the evaluation accuracy by bringing the design closer to reality, at the expense of increased implementation effort and reduced flexibility. Finally, it is also important that testing and verification must be considered to ensure correctness.

2) Proposed Approaches: Several concrete NoC architectures have been presented in the literature. In [2], the authors present the SPIN (scalable, packet switched, on-chip micro-network) architecture and implement a 32-port network architecture using a 0.13-μm process. This architecture uses credit-based flow control to provide QoS.

A flexible field-programmable gate array (FPGA)-based NoC design that consists of processors and reconfigurable components is presented in [11]. The FPGA prototype presented in [121] illustrates the impact of application-specific long-range links on the performance and energy consumption of 2-D mesh networks. The adaptive system-on-chip (aSoC) architecture presented in [97] supports compile-time scheduling for on-chip communication and provides software-based dynamic routing. The RAW chip [163] attacks the wire-delay problem by proposing a direct software interface to the physical resources. The static network used in the RAW chip also enables new application domains.

The 80-core teraflop chip recently introduced by Intel [166] is a good example of an aggressive NoC prototyping effort. The chip uses a 2-D mesh with mesochronous clocking for a high-bandwidth scalable design. The authors in [93] present a highly optimized NoC implementation using hierarchical star topology. Finally, the work presented in [79] addresses both architectural aspects and circuit-level techniques for practical NoC implementation.

We note, however, that most studies dealing with concrete NoC implementations lack performance evaluation under *real* driver applications. This is an important issue that needs to be addressed in order to bring the NoC prototypes closer to real applications. Toward this end, the authors in [91] compare and contrast the NoC, bus-, and P2P-based implementations of an MPEG-2 encoder using an FPGA-based prototype. The advantages of the NoC approach are illustrated in terms of scalability, throughput, energy consumption, and area, both analytically and using direct measurements on the prototype.

In NoCs, the routers and links have been utilized to test the PEs and the network itself based on built-in self-test mechanisms [4], [55], [65], [99]. In [4], the authors propose a scalable test strategy for the routers in a NoC based on partial scan and an IEEE 1500-compliant test wrapper. Similarly, the strategy proposed in [65] exploits the regularity of the switches and broadcasts the test vectors through the minimum spanning tree to test the switches concurrently. The authors in [55] propose testing the routing logic and FIFO buffers recursively by utilizing the NoC component that already passed the test. The work presented in [99] also considers the power consumption required for testing purposes.

NoC verification has received less attention compared to other design aspects or even testing. The NoC verification approach in [53] relies on monitoring the network traffic and checking special events such as connection opened/closed, data received by a connection, etc. The authors in [142] present a formal verification approach for asynchronous architectures. The proposed approach is then illustrated using the asynchronous

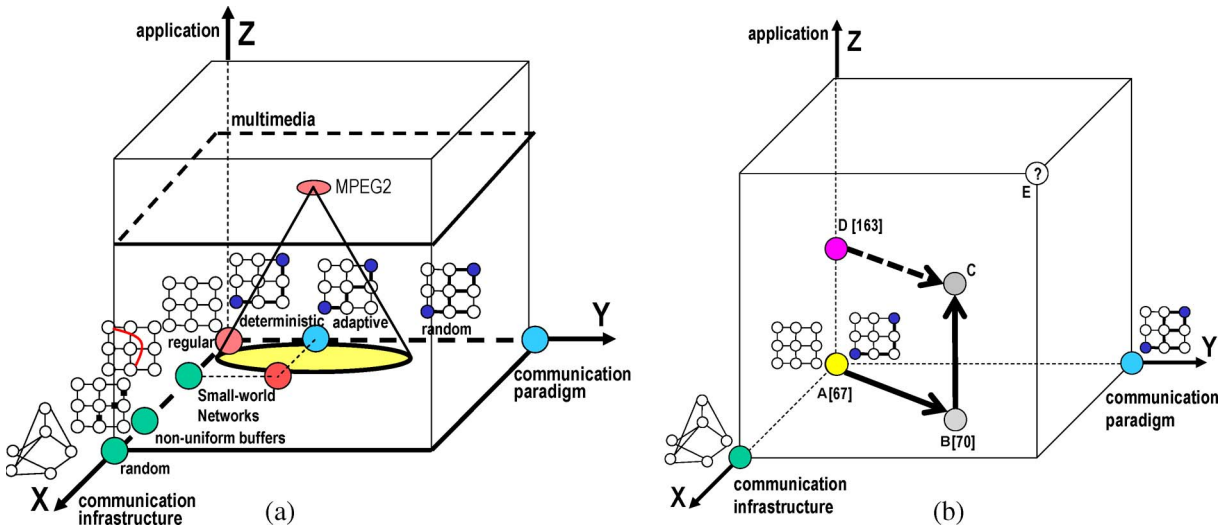


Fig. 3. (a) NoC design space in a 3-D representation. As an example, optimizing the communication infrastructure (Section V) in conjunction with the communication paradigm (Section IV) covers a region in the $x - y$ plane. (b) Mapping a few approaches to this 3-D design space and visualization of possible trajectories of research ideas. Future research in NoC design will likely cover this 3-D representation as densely as possible.

NoC implementation in [13]. Finally, the work in [125] presents the MAIA framework, a low level verification language, which is used for automated NoC generation, automated production of NoC–IP core interfaces, and seamless analysis of NoC traffic parameters.

3) Open Problems: Testing and verification techniques are indispensable components of NoC design flows. Therefore, NoC design methodologies need to be complemented by efficient mechanisms for testing the NoC communication fabric and the functional cores. Pin-count limitations restrict the use of I/O pins dedicated for testing; therefore, novel approaches are needed for the testing and verification of NoCs. While there has been increased attention for testing and verification techniques, more work in this direction is clearly needed.

VII. Interaction Among Various Research Problems

Having discussed all these issues in detail, in this section, we consider the interaction among them. Such interactions can be seen in Fig. 3 by adopting again the 3-D view in Section II on the design space. In this representation, the level of randomness increases toward the tips of the x-, y-, and z-arrows.

The *communication infrastructure* is shown along the x-axis in Fig. 3(a). This dimension defines how nodes are interconnected to each other and reflects the properties of the underlying network. As shown in Fig. 3(a), examples include regular networks, small-world networks, networks with customized buffer sizes, or fully customized networks. Problems P9–P13 discussed in Section V fall along this dimension. The *communication paradigm* is shown by the y-axis Fig. 3(b). This dimension captures the dynamics of transferring packets (e.g., deterministic versus adaptive routing, QoS-based routing, etc.) through the network. Problems P4–P8 discussed in Section IV fall along this dimension. Finally, the third dimension, *application mapping*, defines how different tasks of the target application get mapped to the network nodes (e.g., how various traffic sources and sinks are distributed across the network) in order to satisfy various design constraints. Problems P1–P3 fall along this dimension. Although not shown in Fig. 3, analysis, simulation, and validation (Section VI, P14–15) can be viewed as encompassing the entire design space; these steps are an integral part of work in any dimension and for research spanning multiple dimensions.

It is important to note that each axis represents a *continuum* rather than a discrete set of solutions. Also, any point inside the 3-D representation in Fig. 3 actually denotes a possible design tradeoff among various problems and techniques already discussed. For example, designing a routing algorithm while solving the topology customization problem corresponds to finding a legitimate point in the x–y plane. More generally, the shaded circle within the x–y plane shows a *range* of design solutions which correspond to various design tradeoffs between the communication infrastructure and the communication paradigm. Indeed, by projecting a particular application [e.g., MPEG-2 in Fig. 3(a)] onto the x–y plane, one can see that any solution within the shaded circle can implement the application at hand while pursuing different tradeoffs. By the same token, by projecting up a particular point from the x–y plane onto the application plane, we can see a family of related multimedia applications (e.g., MPEG-2 video and audio) benefiting from such a particular implementation.

Interestingly enough, this 3-D representation also allows us to map various approaches proposed to date in this design space. For instance, the approach in [67] can be placed close to the origin in Fig. 3(b) (see point A) because this addresses the mapping problem for application specific NoCs, the topology is completely regular, and routing is XY. By generalizing this approach to accommodate adaptive routing and regions of irregular size as suggested in [70], the representation moves from point A to B (see Fig. 3(b)[3]). Future work can perhaps generalize this idea even further to allow for 3-D communication and multiple applications that run simultaneously; this

[3]Of course, similar positions within the XY plane would correspond to other approaches, for instance [60] or [134].

would correspond to reaching point C in Fig. 3(b). Similarly, an approach like that in [163] [which is placed at point D in Fig. 3(b)] can also reach point C if generalized to allow for 3-D communication and some form of adaptive routing.

From this discussion, one can see that this 3-D representation can be used to visualize the trajectories that various ideas undertake, compare, and contrast proposed approaches or, better yet, identify important "bald spots" where groundbreaking research is needed. We anticipate that future research in NoC design will likely cover this 3-D representation as densely as possible. In other words, more complex design techniques, together with software and application programming, will play an increasingly important role in NoC research. This will enable the design and optimization of reconfigurable platforms that can accommodate multi-application scenarios and exploit various power/performance tradeoffs at run time. The long-term hope is for further research solutions that enable networks with properties in the vicinity of point E in Fig. 3. So far, only non-engineered networks (e.g., human networks involved in spreading of airborne infectious diseases, online communities, etc.) exhibit such powerful capabilities.

VIII. Conclusion

In this paper, we have discussed several research challenges in the design of NoCs. Rather than simply enumerating the relevant prior work, we have provided a framework for discussing each problem including motivation, formulation, and open research issues. While this paper focuses on the circuit-, microarchitectural-, and system-level issues of the communication infrastructure, future work can cover the programmability and other software issues using a similar philosophy.

Acknowledgment

The authors would also like to thank Dr. M. Kishinevsky of Intel, Dr. J. Hu of Tabula Inc., and Dr. P. Bogdan, Dr. S. Garg, and Dr. D. Marculescu of Carnegie Mellon University for the valuable feedback in the early stages of the manuscript.

References

[1] P. Abad, V. Puente, J. A. Gregorio, and P. Prieto, "Rotary router: An efficient architecture for CMP interconnection networks," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2007, pp. 116–125.

[2] A. Adriahantenaina and A. Greiner, "Micro-network for SoC: Implementation of a 32-port SPIN network," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2003, pp. 1128–1129.

[3] M. A. Al Faruque, T. Ebi, and J. Henkel, "Run-time adaptive on-chip communication scheme," in *Proc. Int. Conf. Comput.-Aided Des.*, San Jose, CA, 2007, pp. 26–31.

[4] A. M. Amory *et al.*, "A scalable test strategy for network-on-chip routers," in *Proc. IEEE Int. Test Conf.*, Nov. 2005, pp. 591–599.

[5] F. Angiolini, P. Meloni, S. Carta, L. Benini, and L. Raffo, "Contrasting a NoC and a traditional interconnect fabric with layout awareness," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 124–129.

[6] F. Angiolini, D. Atienza, S. Murali, L. Benini, and G. De Micheli, "Reliability support for on-chip memories using networks-on-chip," in *Proc. Int. Conf. Comput. Des.*, Oct. 2006, pp. 389–396.

[7] G. Ascia, V. Catania, and M. Palesi, "Multi-objective mapping for mesh-based NoC architectures," in *Proc. Int. Conf. Hardware-Softw. Codesign Syst. Synthesis*, Sep. 2004, pp. 182–187.

[8] W. Bainbridge and S. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding," in *Proc. Int. Symp. Asynchronous Circuits Syst.*, Mar. 2001, pp. 118–126.

[9] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proc. Int. Conf. Supercomputing*, Jun. 2006, pp. 187–198.

[10] N. Banerjee, P. Vellank, and K. S. Chatha, "A power and performance model for network-on-chip architectures," in *Proc. Des., Autom. Test Eur. Conf.*, Feb. 2004, pp. 1250–1255.

[11] T. A. Bartic *et al.*, "Highly scalable network on chip for reconfigurable systems," in *Proc. Int. Symp. Syst.-on-Chip*, Nov. 2003, pp. 79–82.

[12] P. Beerel and M. E. Roncken, "Low power and energy efficient asynchronous design," *J. Low Power Electron.*, vol. 3, no. 3, pp. 234–253, Dec. 2007.

[13] E. Beigne, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin, "An asynchronous NOC architecture providing low latency service and its multi-level design framework," in *Proc. Int. Symp. Asynchronous Circuits Syst.*, May 2005, pp. 54–63.

[14] E. Beigne, F. Clermidy, S. Miermont, and P. Vivet, "Dynamic voltage and frequency scaling architecture for units integration within a GALS NoC," in *Proc. Int. Symp. Netw. Chip*, 2008, pp. 129–138.

[15] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[16] D. Bertozzi *et al.*, "NoC synthesis flow for customized domain specific multiprocessor systems-on-chip," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 2, pp. 113–129, Feb. 2005.

[17] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: The energy–reliability tradeoff," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 6, pp. 818–831, Jun. 2005.

[18] D. Bertozzi, A. Acquaviva, D. Bertozzi, and A. Poggiali, "Supporting task migration in multi-processor systems-on-chip: A feasibility study," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 1–6.

[19] P. Bhojwani, J. D. Lee, and R. Mahapatra, "SAPP: Scalable and adaptable peak power management in NoCs," in *Proc. Int. Symp. Low Power Electron. Devices*, Aug. 2007, pp. 340–345.

[20] T. Bjerregaard and J. Sparso, "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2005, pp. 1226–1231.

[21] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Comput. Surv.*, vol. 38, no. 1, pp. 1–51, Mar. 2006.

[22] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," Princeton Univ., Princeton, NJ, Tech. Rep. TR-811-08, Jan. 2008.

[23] P. Bogdan, T. Dumitras, and R. Marculescu, "Stochastic communication: A new paradigm for fault-tolerant networks-on-chip," *Hindawi VLSI Design*, Feb. 2007.

[24] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," *J. Syst. Architecture: EUROMICRO J.*, vol. 50, no. 2/3, pp. 105–128, Feb. 2004.

[25] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Routing table minimization for irregular mesh NoCs," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 942–947.

[26] J. W. van den Brand, C. Ciordas, K. Goossens, and T. Basten, "Congestion-controlled best-effort communication for networks-on-chip," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 948–953.

[27] G. Campobello, M. Castano, C. Ciofi, and D. Mangano, "GALS networks on chip: A new solution for asynchronous delay-insensitive links," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 160–165.

[28] L. P. Carloni, K. L. McMillan, and A. L. Sangiovanni-Vincentelli, "Theory of latency-insensitive design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 9, pp. 1059–1076, Sep. 2001.

[29] V. Catania, R. Holsmark, S. Kumar, and M. Palesi, "A methodology for design of application specific deadlock-free routing algorithms for NoC systems," in *Proc. Int. Conf. Hardware-Softw. Codesign Syst. Synthesis*, Oct. 2006, pp. 142–147.

[30] J. Chan and S. Parameswaran, "NoCEE: Energy macro-model extraction methodology for network on chip routers," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2005, pp. 254–259.

[31] S. C. Chan, K. L. Shepard, and P. J. Restle, "Design of resonant global clock distributions," in *Proc. Int. Conf. Comput. Des.*, Oct. 2003, pp. 248–253.

[32] M. F. Chang *et al.*, "CMP network-on-chip overlaid with multi-band RF-interconnect," in *Proc. Int. Symp. High-Performance Comput. Architecture*, Feb. 2008, pp. 191–202.

[33] T. Chelcea and S. M. Nowick, "Robust interfaces for mixed-timing systems with application to latency-insensitive protocols," in *Proc. Des. Autom. Conf.*, Jun. 2001, pp. 21–26.

[34] X. Chen and L. Peh, "Leakage power modeling and optimization in interconnection networks," in *Proc. Int. Symp. Low Power Electron. Des.*, Aug. 2003, pp. 90–95.

[35] C.-L. Chou and R. Marculescu, "Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 10, pp. 1866–1879, Oct. 2008.

[36] M. Colajanni, B. Ciciani, and F. Quaglia, "Performance analysis of wormhole switching with adaptive routing in a two-dimensional torus," in *Proc. Int. Eur.-Par Conf. Parallel Process.*, 1999, vol. 1685, pp. 165–172.

[37] J. Cortadella, M. Kishinevsky, and B. Grundmann, "Synthesis of synchronous elastic architectures," in *Proc. Des., Autom. Conf.*, Jul. 2006, pp. 657–662.

[38] A. K. Coskun, T. S. Rosing, and K. Whisnant, "Temperature aware task scheduling in MPSoCs," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 1659–1664.

[39] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Mateo, CA: Morgan Kaufmann, 2004.

[40] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. Des. Autom. Conf.*, Jun. 2001, pp. 684–689.

[41] W. J. Dally, "Virtual-channel flow control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 3, no. 2, pp. 194–205, Mar. 1992.

[42] J. Dielissen, A. Radulescu, K. Goossens, and E. Rijpkema, "Concepts and implementation of the Philips network-on-chip," *IP-Based SoC Design*, 2003.

[43] R. Dobkin, R. Ginosar, and C. Sotiriou, "Data synchronization issues in GALS SoCs," in *Proc. Int. Symp. Asynchronous Circuits Syst.*, Apr. 2004, pp. 170–179.

[44] J. Draper and J. Ghosh, "A comprehensive analytical model for wormhole routing in multicomputer systems," *J. Parallel Distrib. Comput.*, vol. 23, no. 2, pp. 202–214, Nov. 1994.

[45] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: An Engineering Approach*. San Mateo, CA: Morgan Kaufmann, 2002.

[46] J. Duato *et al.*, "A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks," in *Proc. Int. Symp. High-Performance Comput. Architecture*, Feb. 2005, pp. 108–119.

[47] N. Eisley and L. Peh, "High-level power analysis for on-chip networks," in *Proc. Int. Conf. Compilers, Architectures Synthesis Embedded Syst.*, Sep. 2004, pp. 104–115.

[48] N. Enright-Jerger, L. Peh, and M. Lipasti, "Circuit-switched coherence," in *Proc. Int. Symp. Netw.-on-Chips*, May 2008, pp. 193–202.

[49] N. Enright-Jerger, L.-S. Peh, and M. Lipasti, "Virtual circuit tree multicasting: A case for on-chip hardware multicast support," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2008, pp. 229–240.

[50] A. Ejlali, B. M. Al-Hashimi, P. Rosinger, and S. G. Miremadi, "Joint consideration of fault-tolerance, energy-efficiency and performance in on-chip networks," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 1647–1652.

[51] N. Genko, G. De Micheli, D. Atienza, J. Mendias, R. Hermida, and F. Catthoor, "A complete network-on-chip emulation framework," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2005, pp. 246–251.

[52] C. M. Goldfeder, "Frequency-based code placement for embedded multiprocessors," in *Proc. Des. Autom. Conf.*, Jul. 2005, pp. 696–699.

[53] K. Goossens *et al.*, "A design flow for application-specific networks on chip with guaranteed performance to accelerate SoC design and verification," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2005, pp. 1182–1187.

[54] P. Gratz, C. Kim, R. McDonald, S. W. Keckler, and D. C. Burger, "Implementation and evaluation of on-chip network architectures," in *Proc. Int. Conf. Comput. Des.*, Oct. 2006, pp. 477–484.

[55] C. Grecu, P. P. Pande, B. Wang, A. Ivanov, and R. Saleh, "Methodologies and algorithms for testing switch-based NoC interconnects," in *Proc. Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2005, pp. 238–246.

[56] C. Grecu, A. Ivanov, P. P. Pande, A. Jantsch, E. Salminen, U. Ogras, and R. Marculescu, "Towards open network-on-chip benchmarks," in *Proc. Int. Symp. Netw.-on-Chip*, May 2007, p. 205.

[57] F. Gruian, "Hard real-time scheduling for low-energy using stochastic data and DVS processors," in *Proc. Int. Symp. Low-Power Electron. Des.*, Aug. 2001, pp. 46–51.

[58] Z. Guz, I. Walter, E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Efficient link capacity and QoS design for network-on-chip," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 9–14.

[59] A. Hansson, M. Coenen, and K. Goossens, "Undisrupted quality-of-service during reconfiguration of multiple applications in networks on chip," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 954–959.

[60] A. Hansson, K. Goossens, and A. Radulescu, "A unified approach to mapping and routing on a network-on-chip for both best-effort and guaranteed service traffic," *Hindawi VLSI Design*, May 2007.

[61] M. Harmanci, N. Escudero, Y. Leblebici, and P. Ienne, "Quantitative modeling and comparison of communication schemes to guarantee quality-of-service in networks-on-chip," in *Proc. Int. Symp. Circuits Syst.*, May 2005, pp. 1782–1785.

[62] J. Henkel, W. Wolf, and S. Chakradhar, "On-chip networks: A scalable, communication-centric embedded system design paradigm," in *Proc. VLSI Des.*, Jan. 2004, pp. 845–851.

[63] W. H. Ho and T. M. Pinkston, "A methodology for designing efficient on-chip interconnects on well-behaved communication patterns," in *Proc. Int. Symp. High-Performance Comput. Architecture*, Feb. 2003, pp. 377–388.

[64] M. Horowitz, R. Ho, and K. Mai, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.

[65] M. Hosseinabady, A. Dalirsani, and Z. Navabi, "Using the inter- and intra-switch regularity in NoC switch testing," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 361–366.

[66] P. Hu and L. Kleinrock, "An analytical model for wormhole routing with finite size input buffers," in *Proc. 15th Int. Teletraffic Congr.*, Jun. 1997, pp. 549–560.

[67] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NOC architectures under performance constraints," in *Proc. Asia South Pacific Des. Autom. Conf.*, Kitakyushu, Japan, Jan. 2003, pp. 233–239.

[68] J. Hu and R. Marculescu, "Communication and task scheduling of application-specific networks-on-chip," *Proc. Inst. Elect. Eng.—Comput. Digit. Tech.*, vol. 152, no. 5, pp. 643–651, Sep. 2005.

[69] J. Hu and R. Marculescu, "DyAD—Smart routing for networks-on-chip," in *Proc. Des. Autom. Conf.*, Jun. 2004, pp. 260–263.

[70] J. Hu and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 4, pp. 551–562, Apr. 2005.

[71] J. Hu, U. Y. Ogras, and R. Marculescu, "System-level buffer allocation for application-specific networks-on-chip router design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 12, pp. 2919–2933, Dec. 2006.

[72] W. Hung *et al.*, "Thermal-aware IP virtualization and placement for networks-on-chip architecture," in *Proc. Int. Conf. Comput. Des.*, 2004, pp. 430–437.

[73] K. Z. Ibrahim, "Correlation between detailed and simplified simulations in studying multiprocessor architecture," in *Proc. Int. Conf. Comput. Des.*, Oct. 2005, pp. 387–392.

[74] *Networks-on-Chip*, A. Jantsch and H. Tenhunen, Eds. Norwell, MA: Kluwer, 2003.

[75] A. Jantsch, R. Lauter, and A. Vitkowski, "Power analysis of link level and end-to-end data protection in networks on chip," in *Proc. Int. Symp. Circuits Syst.*, May 2005, pp. 1770–1773.

[76] A. P. Jose, G. Patounakis, and K. L. Shepard, "Near speed-of-light on-chip interconnects using pulsed current-mode signalling," in *Proc. Symp. VLSI Circuits*, Jun. 2005, pp. 108–111.

[77] S. W. Keckler *et al.*, "A wire-delay scalable microprocessor architecture for high performance systems," in *Proc. Solid-State Circuits Conf.*, Feb. 2003, pp. 168–169.

[78] B. Kim and V. Stojanovic, "Equalized interconnects for on-chip networks: Modeling and optimization framework," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2007, pp. 552–559.

[79] D. Kim, K. Kim, J. Kim, S. Lee, and H. Yoo, "Solutions for real chip implementation issues of NoC and their application to memory-centric NoC," in *Proc. Int. Symp. Netw.-on-Chips*, May 2007, pp. 30–39.

[80] E. J. Kim *et al.*, "Energy optimization techniques in cluster interconnects," in *Proc. Int. Symp. Low Power Electron. Des.*, Aug. 2003, pp. 459–464.

[81] M. Kim, D. Kim, and G. E. Sobelman, "Adaptive scheduling for CDMA-based networks-on-chip," in *Proc. IEEE Northeast Workshop Circuits Syst.*, May 2005, pp. 357–360.

[82] J. S. Kim, M. B. Taylor, J. Miller, and D. Wentzlaff, "Energy characterization of a tiled architecture processor with on-chip networks," in *Proc. Int. Symp. Low Power Electron. Des.*, Aug. 2003, pp. 424–427.

[83] J. Kim, D. Park, C. Nicopoulos, V. Narayanan, and C. Das, "Design and analysis of an NoC architecture from performance, reliability and energy perspective," in *Proc. 1st Symp. Architectures Netw. Commun. Syst.*, Oct. 2005, pp. 173–182.

[84] J. Kim, C. A. Nicopoulos, D. Park, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "A gracefully degrading and energy-efficient modular router architecture for on-chip networks," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2006, pp. 4–15.

[85] J. Kim *et al.*, "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2007, pp. 138–149.
[86] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: A cost-efficient topology for high-radix networks," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2007, pp. 126–137.
[87] T. Kogel *et al.*, "A modular simulation framework for architectural exploration of on-chip interconnection networks," in *Proc. Int. Conf. Hardware-Software Codesign Syst. Synthesis*, Oct. 2003, pp. 7–12.
[88] A. Kumar, L. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2007, pp. 150–161.
[89] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2005, pp. 408–419.
[90] K. Lahiri *et al.*, "Evaluation of the traffic-performance characteristics of system-on-chip communication architectures," in *Proc. Int. Conf. VLSI Des.*, Oct. 2000, pp. 29–35.
[91] H. G. Lee, N. Chang, U. Y. Ogras, and R. Marculescu, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus and network-on-chip approaches," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 12, no. 3, pp. 1–20, Aug. 2007.
[92] J. W. Lee, A. Ng, and K. Asanovic, "Globally-synchronized frames for guaranteed quality-of-service in on-chip networks," in *Proc. Int. Symp. Comput. Architecture*, 2008, pp. 89–100.
[93] K. Lee *et al.*, "A 51 mW 1.6 GHz on-chip network for low-power heterogeneous SoC platform," in *Proc. Int. Solid-State Circuits Conf.*, Feb. 2004, pp. 152–518.
[94] T. Lehtonen, P. Liljeberg, and J. Plosila, "Fault tolerance analysis of NoC architectures," in *Proc. Int. Symp. Circuits Syst.*, May 2007, pp. 361–364.
[95] L. F. Leung and C. Y. Tsui, "Optimal link scheduling on improving best-effort and guaranteed services performance in network-on-chip systems," in *Proc. Des. Autom. Conf.*, Jul. 2006, pp. 833–838.
[96] F. Li, G. Chen, and M. Kandemir, "Compiler-directed voltage scaling on communication links for reducing power consumption," in *Proc. Int. Conf. Comput.-Aided Des.*, 2005, pp. 456–460.
[97] J. Liang, A. Laffely, S. Srinivasan, and R. Tessier, "An architecture and compiler for scalable on-chip communication," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 7, pp. 711–726, Jul. 2004.
[98] T. Lin and L. T. Pileggi, "Throughput-driven IC communication fabric synthesis," in *Proc. Int. Conf. Comput.-Aided Des.*, 2002, pp. 274–279.
[99] C. Liu, J. Shi, E. Cota, and V. Iyengar, "Power-aware test scheduling in network-on-chip using variable-rate on-chip clocking," in *Proc. VLSI Test Symp.*, May 2005, pp. 349–354.
[100] Z. Lu, M. Liu, and A. Jantsch, "Layered switching for networks on chip," in *Proc. Des. Autom. Conf.*, Jun. 2007, pp. 122–127.
[101] J. Luo and N. K. Jha, "Power-conscious joint scheduling of periodic task graphs and aperiodic tasks in distributed real-time embedded systems," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2000, pp. 357–364.
[102] T. S. Mak, P. Sedcole, P. Y. Cheung, W. Luk, and K. P. Lam, "A hybrid analog–digital routing network for NoC dynamic routing," in *Proc. Int. Symp. Netw.-on-Chip*, May 2007, pp. 173–182.
[103] J. Madsen, S. Mahadevan, K. Virk, and M. Gonzales, "Network-on-chip modeling for system-level multiprocessor simulation," in *Proc. IEEE Int. Real-Time Syst. Symp.*, Dec. 2003, pp. 265–274.
[104] S. Mahadevan *et al.*, "A network traffic generator model for fast network-on-chip simulation," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2005, pp. 780–785.
[105] S. Manolache, P. Eles, and Z. Peng, "Fault and energy-aware communication mapping with guaranteed latency for applications implemented on NoC," in *Proc. Des. Autom. Conf.*, Jul. 2005, pp. 266–269.
[106] T. Marescaux and H. Corporaal, "Introducing the superGT network-on-chip," in *Proc. Des. Autom. Conf.*, Jun. 2007, pp. 116–121.
[107] *Networks on Chips: Technology and Tools*, G. De Micheli and L. Benini, Eds. San Mateo, CA: Morgan Kaufmann, 2006.
[108] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip," in *Proc. Des., Autom. Test Eur. Conf.*, Feb. 2004, pp. 890–895.
[109] R. Mishra *et al.*, "Energy aware scheduling for distributed real-time systems," in *Proc. Int. Parallel Distrib. Process. Symp.*, Apr. 2003, p. 12b.
[110] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2004, pp. 188–197.
[111] S. Murali, D. Atienza, L. Benini, and G. De Micheli, "A method for routing packets across multiple paths in NoCs with in-order delivery and fault-tolerance guarantees," *Hindawi VLSI Design*, May 2007.
[112] S. Murali *et al.*, "Analysis of error recovery schemes for networks on chip," *IEEE Des. Test. Comput.*, vol. 22, no. 5, pp. 434–442, Sep./Oct. 2005.
[113] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G. De Micheli, "A methodology for mapping multiple use-cases onto networks on chips," in *Proc. Des. Autom. Test Eur. Conf.*, Mar. 2006, pp. 118–123.
[114] S. Murali *et al.*, "Designing application-specific networks on chips with floorplan information," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2006, pp. 355–362.
[115] S. Murali and G. De Micheli, "Bandwidth-constrained mapping of cores onto NoC architectures," in *Proc. Des., Autom. Test Eur. Conf.*, Feb. 2004, pp. 896–901.
[116] C. A. Nicopoulos *et al.*, "ViChaR: A dynamic virtual channel regulator for network-on-chip routers," in *Proc. Int. Symp. Microarchitecture*, Dec. 2006, pp. 333–346.
[117] E. Nigussie *et al.*, "High-performance long NoC link using delay-insensitive current-mode signaling," *Hindawi VLSI Design—Special Issue on Networks-on-Chip*, vol. 2007, Mar. 2007.
[118] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2003, pp. 1126–1127.
[119] U. Y. Ogras and R. Marculescu, "Energy- and performance-driven NoC communication architecture synthesis using a decomposition approach," in *Proc. Des., Autom. Test Eur.*, Mar. 2005, pp. 352–357.
[120] U. Y. Ogras, J. Hu, and R. Marculescu, "Key research problems in NoC design: A holistic perspective," in *Proc. Int. Conf. Hardware-Softw. Codesign Syst. Synthesis*, Sep. 2005, pp. 69–74.
[121] U. Y. Ogras and R. Marculescu, "'It's a small world after all': NoC performance optimization via long-range link insertion," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.—Special Section Hardware/Softw. Codesign Syst. Synthesis*, vol. 14, no. 7, pp. 693–706, Jul. 2006.
[122] U. Y. Ogras and R. Marculescu, "Analysis and optimization of prediction-based flow control in networks-on-chip," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 13, no. 1, pp. 1–28, Jan. 2008.
[123] U. Y. Ogras and R. Marculescu, "Analytical router modeling for networks-on-chip performance analysis," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 1096–1101.
[124] U. Y. Ogras, R. Marculescu, D. Marculescu, and E. G. Jung, "Design and management of voltage–frequency island partitioned networks-on-chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.—Special Section Networks-on-Chip*, Feb./Mar. 2009. to appear.
[125] L. Ost, A. Mello, J. Palma, F. Moraes, and N. Calazans, "MAIA: A framework for networks on chip generation and verification," in *Proc. Asia South Pacific Des. Autom. Conf.*, Jan. 2005, pp. 49–52.
[126] G. Palermo and C. Silvano, "PIRATE: A framework for power/performance exploration of network-on-chip architectures," in *Proc. Int. Workshop Power Timing Model., Optimization Simul.*, Sep. 2004, pp. 521–531.
[127] D. Pamunuwa *et al.*, "Layout, performance and power trade-offs in mesh-based network-on-chip architectures," in *Proc. IFIP Int. Conf. Very Large Scale Integr.*, Dec. 2003, p. 362.
[128] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Trans. Comput.*, vol. 54, no. 8, pp. 1025–1040, Aug. 2005.
[129] P. M. Pardalos, F. Rendl, and H. Wolkowicz, "The quadratic assignment problem: A survey and recent developments," in *Proc. DIMACS Workshop*, P. Pardalos and H. Wolkowicz, Eds., *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 1994, vol. 16, pp. 1–42.
[130] K. Park and W. Willinger, "Self-similar network traffic: An overview," in *Self-Similar Network Traffic and Performance Evaluation*, K. Park and W. Willinger, Eds. New York: Wiley-Interscience, 1999.
[131] C. S. Patel, S. M. Chai, S. Yalamanchili, and D. E. Schimmel, "Power constrained design of multiprocessor interconnection networks," in *Proc. Int. Conf. Comput. Des.*, Oct. 1997, pp. 408–416.
[132] L. Peh and W. J. Dally, "Flit-reservation flow control," in *Proc. Int. Symp. High-Performance Comput. Architecture*, Jan. 2000, pp. 73–84.
[133] L. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *Proc. Int. Symp. High-Performance Comput. Architecture*, Jan. 2001, pp. 255–266.
[134] A. Pinto, L. P. Carloni, and A. L. Sangiovanni-Vincentelli, "Efficient synthesis of networks on chip," in *Proc. Int. Conf. Comput. Des.*, Oct. 2003, pp. 146–150.

[135] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Fault tolerant algorithms for network-on-chip interconnect," in *Proc. IEEE Symp. VLSI*, Feb. 2004, pp. 46–51.

[136] D. Pham *et al.*, "The design and implementation of a first-generation CELL processor," in *Proc. Solid-State Circuits Conf.*, Feb. 2005, pp. 184–592.

[137] P. Pop *et al.*, "An approach to incremental design of distributed embedded systems," in *Proc. Des. Autom. Conf.*, Jun. 2001, pp. 450–455.

[138] P. Poplavko, T. Basten, M. Bekooij, J. van Meerbergen, and B. Mesman, "Task-level timing models for guaranteed performance in multiprocessor networks-on-chip," in *Proc. Int. Conf. Compilers, Architecture Synthesis Embedded Syst.*, 2003, pp. 63–72.

[139] V. Puente, J. A. Gregorio, F. Vallejo, and R. Beivide, "Immunet: A cheap and robust fault-tolerant packet routing mechanism," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2004, pp. 198–209.

[140] A. Pullini, F. Angiolini, D. Bertozzi, and L. Benini, "Fault tolerance overhead in network-on-chip flow control schemes," in *Proc. Symp. Integr. Circuits Syst. Des.*, Sep. 2005, pp. 224–229.

[141] I. Saastamoinen, M. Alho, and J. Nurmi, "Buffer implementation for proteo network-on-chip," in *Proc. Int. Symp. Circuits Syst.*, May 2003, pp. II-113–II-116.

[142] G. Salaun, W. Serwe, Y. Thonnart, and P. Vivet, "Formal verification of CHP specifications with CADP illustration on an asynchronous network-on-chip," in *Proc. IEEE Int. Symp. Asynchronous Circuits Syst.*, 2007, pp. 73–82.

[143] M. T. Schmitz, B. M. Al-Hashimi, and P. Eles, "Iterative schedule optimization for voltage scalable distributed embedded systems," *ACM Trans. Embedd. Comput. Syst.*, vol. 3, no. 1, pp. 182–217, Feb. 2004. DOI=http://doi.acm.org/10.1145/972627.972636.

[144] D. Seo, A. Ali, W. Lim, N. Rafique, and M. Thottethodi, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," in *Proc. Int. Symp. Comput. Architecture*, Jun. 2005, pp. 432–443.

[145] L. Shang, L. Peh, and N. K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *Proc. Int. Symp. High-Performance Comput. Architecture*, Jan. 2003, pp. 91–102.

[146] L. Shang, L. Peh, A. Kumar, and N. K. Jha, "Thermal modeling, characterization and management of on-chip networks," in *Proc. Int. Symp. Microarchitecture*, Dec. 2004, pp. 67–78.

[147] A. Scherrer, A. Fraboulet, and T. Risset, "Automatic phase detection for stochastic on-chip traffic generation," in *Proc. Int. Conf. Hardware-Softw. Codesign*, Oct. 2006, pp. 88–93.

[148] A. Sheibanyrad, I. M. Panades, and A. Greiner, "Systematic comparison between the asynchronous and the multi-synchronous implementations of a network-on-chip architecture," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 1090–1095.

[149] A. Shibayama, K. Nose, S. Torii, M. Mizuno, and M. Edahiro, "Skew-tolerant global synchronization based on periodically all-in-phase clocking for multi-core SOC platforms," in *Proc. Symp. VLSI Circuits*, Jun. 2007, pp. 158–159.

[150] B. Shim and N. R. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 4, pp. 336–348, Apr. 2006.

[151] D. Shin and J. Kim, "Power-aware communication optimization for networks-on-chips with voltage scalable links," in *Proc. Int. Conf. Hardware/Softw. Codesign Syst. Synthesis*, Sep. 2004, pp. 170–175.

[152] D. Shin, J. Kim, and S. Lee, "Intra-task voltage scheduling for low-energy hard real-time applications," *IEEE Des. Test. Comput.*, vol. 18, no. 2, pp. 20–30, Mar./Apr. 2001.

[153] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proc. Int. Conf. Dependable Syst. Netw.*, Jun. 2002, pp. 389–398.

[154] T. Simunic and S. Boyd, "Managing power consumption in networks on chips," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2002, pp. 110–116.

[155] V. Soteriou and L. Peh, "Design space exploration of power-aware on/off interconnection networks," in *Proc. Int. Conf. Comput. Des.*, Oct. 2004, pp. 510–517.

[156] V. Soteriou, H.-S. Wang, and L. Peh, "A statistical traffic model for on-chip interconnection networks," in *Proc. Int. Symp. Model., Anal., Simul. Comput. Telecommun. Syst.*, Sep. 2006, pp. 104–116.

[157] T. Sparso, M. B. Stensgaard, and J. Sparso, "A scalable, timing-safe, network-on-chip architecture with an integrated clock distribution method," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 648–653.

[158] K. Srinivasan and K. S. Chatha, "A technique for low energy mapping and routing in network-on-chip architectures," in *Proc. Int. Symp. Low Power Electron. Des.*, Aug. 2005, pp. 387–392.

[159] K. Srinivasan, K. S. Chatha, and G. Konjevod, "Linear programming based techniques for synthesis of network-on-chip architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 4, pp. 407–420, Apr. 2006.

[160] K. Srinivasan and K. S. Chatha, "A low complexity heuristic for design of custom network-on-chip architectures," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 130–135.

[161] S. Stuijk, T. Basten, M. Geilen, A. H. Ghamarian, and B. Theelen, "Resource-efficient routing and scheduling of time-constrained streaming communication on networks-on-chip," *J. Syst. Architecture: EUROMICRO J.*, vol. 54, no. 3/4, pp. 411–426, Mar. 2008.

[162] C. Sun, L. Shang, and R. P. Dick, "Three-dimensional multi-processor system-on-chip thermal optimization," in *Proc. Int. Conf. Hardware/Softw. Codesign Syst. Synthesis*, Oct. 2007, pp. 117–122.

[163] M. B. Taylor *et al.*, "The RAW microprocessor: A computational fabric for software circuits and general-purpose programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, Mar./Apr. 2002.

[164] M. B. Taylor, W. Lee, S. Amarasinghe, and A. Agarwal, "Scalar operand networks," *IEEE Trans. Parallel Distrib. Syst.—Special Issue On-chip Netw.*, vol. 16, no. 2, pp. 145–162, Feb. 2005.

[165] B. Towles and W. J. Dally, "Worst-case traffic for oblivious routing functions," in *Proc. ACM Symp. Parallel Algorithms Architectures*, Aug. 2002, pp. 1–8.

[166] S. Vangal *et al.*, "An 80-tile 1.28TFLOPS network-on-chip in 65 nm CMOS," in *Proc. Solid-State Circuits Conf.*, Feb. 2007, pp. 98–589.

[167] G. Varatkar and R. Marculescu, "On-chip traffic modeling and synthesis for MPEG-2 video applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 1, pp. 108–119, Jan. 2004.

[168] G. Varatkar, S. Narayanan, N. R. Shanbhag, and D. L. Jones, "Trends in energy-efficiency and robustness using stochastic sensor network-on-a-chip," in *Proc. ACM Great Lakes Symp. VLSI*, May 2008, pp. 351–354.

[169] G. Varatkar and R. Marculescu, "Communication-aware task scheduling and voltage selection for total systems energy minimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, San Jose, CA, Nov. 2003, pp. 510–517.

[170] H. Wang, L. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," in *Proc. Int. Symp. Microarchitecture*, Nov. 2003, pp. 105–116.

[171] H. Wang, L. Peh, and S. Malik, "A technology-aware and energy-oriented topology exploration for on-chip networks," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2005, pp. 1238–1243.

[172] H. Wang, X. Zhu, L. Peh, and S. Malik, "Orion: A power-performance simulator for interconnection networks," in *Proc. Int. Symp. Microarchitecture*, Nov. 2002, pp. 294–305.

[173] D. Wiklund and D. Liu, "SoCBus: Switched network on chip for hard real time embedded systems," in *Proc. Int. Parallel Distrib. Process. Symp.*, Apr. 2003, p. 78.1.

[174] P. T. Wolkotte, G. J. Smit, G. K. Rauwerda, and L. T. Smit, "An energy efficient reconfigurable circuit-switched network-on-chip," in *Proc. Int. Parallel Distrib. Process. Symp.*, Apr. 2005, p. 155a.

[175] F. Worm, P. Ienne, P. Thiran, and G. D. Micheli, "A robust self-calibrating transmission scheme for on-chip networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 12, pp. 1360–1373, Dec. 2004.

[176] Y. Xie and W. Wolf, "Allocation and scheduling of conditional task graph in hardware/software co-synthesis," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2001, pp. 620–625.

[177] S. Yan and B. Lin, "Design of application-specific 3D networks-on-chip architectures," in *Proc 26th Int. Conf. Comput. Des.*, Lake Tahoe, CA, Oct. 12–15, 2008, pp. 142–149.

[178] T. T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Proc. Des. Autom. Conf.*, Jun. 2002, pp. 524–529.

[179] T. T. Ye and G. De Micheli, "Physical planning for multiprocessor networks and switch fabrics," in *Proc. Int. Conf. Appl.-Specific Syst., Architectures Processors*, 2003, pp. 97–107.

[180] Z. Yu and B. Baas, "Implementing tile-based chip multiprocessors with GALS clocking styles," in *Proc. Int. Conf. Comput. Des.*, Oct. 2006, pp. 174–179.

[181] D. Zhao and Y. Wang, "SD-MAC: Design and synthesis of a hardware-efficient collision-free QoS-aware MAC protocol for wireless network-on-chip," *IEEE Trans. Comput.*, vol. 57, no. 9, pp. 1230–1245, Sep. 2008.

Radu Marculescu (S'94–M'98–SM'07) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1998.

He is currently a Professor with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA. His current research focuses on developing design methodologies and software tools for system-on-chip design, on-chip communication, and ambient intelligence.

Dr. Marculescu is the recipient of the 2000 National Science Foundation's CAREER Award, the 2005 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS Best Paper Award from the IEEE Circuits and Systems Society, and a few Best Paper Awards from major conferences in design automation.

Umit Y. Ogras (S'00) received the Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, in 2007.

He is currently a Research Scientist with Strategic CAD Laboratories, Intel Corporation, Hillsboro, OR. His research interests are in the areas of embedded systems and electronic design automation. In particular, his research focuses on communication-centric design methodologies for nanoscale systems-on-chip, with a special interest on network-on-chip communication architectures.

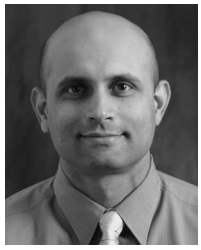
Li-Shiuan Peh received the B.S. degree in computer science from the National University of Singapore, Singapore, in 1995 and the Ph.D. degree in computer science from Stanford University, Stanford, CA, in 2001.

Since 2002, she has been with Princeton University, Princeton, NJ, where she is currently an Associate Professor of electrical engineering with the Department of Electrical Engineering. Her research focuses on low-power interconnection networks, on-chip networks, and parallel computer architectures, and is funded by several grants from the National Science Foundation, the Defense Advanced Research Projects Agency Microelectronics Advanced Research Corporation Gigascale Systems Research Center, and Interconnect Focus Center as well as Intel Corporation.

Dr. Peh is the recipient of the 2003 National Science Foundation's CAREER Award, the Sloan Research Fellowship in 2006, and the 2007 Computing Research Association Anita Borg Early Career Award.

Natalie Enright Jerger received the B.S. degree in computer engineering from Purdue University, West Lafayette, IN, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, in 2008.

She is currently an Assistant Professor in computer engineering with the University of Toronto, Toronto, ON, Canada. Her research interests include communication architectures, cache coherence protocols, and virtual machines for many-core architectures.



Yatin Hoskote (M'96–SM'07) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, and the M.S. and Ph.D. degrees in computer engineering from the University of Texas, Austin.

Since 2000, he has been with the Corporate Technology Group, Intel Corporation, Hillsboro, OR, where he is currently a Principal Engineer.

Dr. Hoskote is the recipient of the 2006 Intel Achievement Award. He is currently enrolled in the Sloan Fellows Program in Innovation and Global Leadership, Massachusetts Institute of Technology, Cambridge.